IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Practical Survey on Visual Odometry for Autonomous Driving in Challenging Scenarios and Conditions

LUCAS R. AGOSTINHO[1*], NUNO M. RICARDO[2*], MARIA I. PEREIRA[3], ANTOINE HIOLLE[4], ANDRY M. PINTO[5]

[1,2,5]Department of Electrical and Computer Engineering, FEUP, Porto, Portugal
[3]Center of Robotics and Autonomous Systems (CRAS), INESC TEC, Porto, Portugal
[4]Bosch Car Multimedia, Braga, Portugal
[*]Corresponding authors: Lucas Agostinho (up201706909@edu.fe.up.pt), Nuno Ricardo (up201706995@edu.fe.up.pt)

This work is funded by the European Commission under the European Union's Horizon 2020 - The EU Framework Programme for Research and Innovation 2014- 2020, under grant agreement No. 871571 (ATLANTIS). It was also supported by European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project nº 047264; Funding Reference: POCI-01-0247-FEDER-047264].

ABSTRACT The expansion of autonomous driving operations requires the research and development of accurate and reliable self-localization approaches. These include visual odometry methods, in which accuracy is potentially superior to GNSS-based traditional techniques while also working in signal-denied areas. This paper presents an in-depth review of state-of-the-art methods in visual and point cloud odometry, along with a direct performance comparison of some of these techniques in the autonomous driving context. The evaluated methods include camera, LiDAR, and multi-modal approaches, featuring knowledge and learning-based algorithms. This set was subject to a series of tests on road driving public datasets, from which the performance of these techniques is benchmarked and quantitatively compared. Furthermore, we closely discuss their effectiveness against challenging conditions such as pronounced lighting variations, open spaces, and the presence of dynamic objects in the scene, grouped by categories. The research addresses and corroborates some of the most prominent limitations of state-of-the-art techniques for visual odometry based on 2D and 3D sensors and points out the stagnation, in terms of performance, of the most recent advances in this area, especially in complex environments. We also examine how multi-modal architectures can circumvent these weaknesses and how the current advances in AI constitute a way to overcome the current stagnation, indexing some opportunities for future research.

INDEX TERMS Visual Odometry, Point Cloud Odometry, Multi-modal Odometry, Ego-motion, Autonomous Driving, Benchmark

I. INTRODUCTION

In the last decade, autonomous driving has been a topic involved in much technical and scientific research. Its numerous benefits, such as increased passenger safety, comfort and convenience, better traffic flow, crewless transport, and reduced fuel consumption, attract investment from large manufacturers responsible for technological advances in autonomous vehicles. For any autonomous mobile agent, the ability to self-locate is essential in every navigation task. Although GNSS (Global Navigation Satellite System) receivers are usually the primary source of self-localization in modern vehicles, mass-market devices provide levels of accuracy and reliability well below those required for use in autonomous vehicles. For this reason, autonomous mobile agents may only loosely depend on satellite data because of the uncertainty of positioning error, signal delays, and quality-of-service issues [1]. This problem further aggravates in urban scenarios due to limited satellite visibility, multipath effect, interferences, and other impairments [1]. All these limitations are compensated by drivers' own visual perception capabilities. In the same way, autonomous cars can be equipped with sensors that provide a similar level of precise relative localization when utilized with proper odometry techniques.

Odometry can be defined as the use of local sensors' data to estimate an agent's change in pose over time, given a

particular starting point. Usually, these methods try to recover the position and orientation of the agent by relying on sensors such as wheel encoders, RAdio Detection And Ranging (RADAR), Inertial Measurement Units (IMUs), cameras, and Light Detection And Ranging (LiDAR), which are becoming more prevalent in modern vehicles. It is also important to acknowledge that these types are not restrictive, as odometry methods can be multi-modal, i.e., different sensors can be used together by a single algorithm. Visual and point cloud-based odometry are emerging as critical methodologies, as the use of cameras and LiDARs is becoming more prevalent in modern vehicles. Unlike GNSS, these sensors do not require external signals to operate. Furthermore, these techniques are much more robust than wheel odometry and easily complemented by IMUs, or GPS [2].

As the demand from government agencies for driver assistance and autonomous safety features increases, the research in related areas of autonomous driving proliferates. Odometry, which takes part in the perception field, is critical to developing such systems. This research provides a general overview of visual, point cloud-based, and multi-modal odometry and compares these categories to a common ground, while considering practical results obtained in the exact same conditions. Furthermore, the rise in Deep Learning (DL) techniques creates the necessity to assess their current state of development in regard to conventional approaches. Another motivation behind this work involved creating an unbiased benchmark of state-of-the-art visual, point cloud-based, and multi-modal approaches using a well-known dataset (KITTI-360 [3]). The benchmark tests different algorithms in challenging situations to validate the reviewed techniques' strengths and limitations. This document also discusses some current issues that could help researchers surpass common visual odometry limitations, such as harsh weather conditions, computational power constraints, and the presence of dynamic objects.

The main contributions of this surveying and benchmarking document are:

- Categorization and theoretical discussion of relevant and promising works in visual odometry, from feature and appearance-based techniques to recent works that leverage the power of Deep Learning; in point cloud-based odometry, including knowledge and learning-based approaches; and in multi-modal odometry, while also analyzing the different types of sensor fusion. These techniques were analyzed using a common evaluation procedure, across the same scenarios;
- Extensive experiments for benchmarking several open-source algorithms with a particular focus on challenging situations such as dynamic environments, open spaces, brightness variations, dense vegetation, turnaround maneuvers, and high velocities;
- Identification of the current challenges for ego-motion estimation, such as the dependency on the scene's appearance, high computational loads, and the presence of moving objects. Analysis and quantification of such conditions on the performance of the different types of addressed methodologies;
- Indexing the current state-of-the-art while providing insight into the present landscape of deep versus knowledge-based approaches and multi-modal architectures and how future research may surpass current results.

This document is organized as follows: sections II and III present the working principles and state-of-the-art works in visual and point cloud-based odometry, respectively. Section IV briefly explains the concepts of data fusion and shows some multi-modal odometry approaches. Section V discusses the results on the already established KITTI odometry benchmark, while section VI shows the results of various open-source methods on selected sequences of the KITTI-360 dataset along with a discussion and comparison of the various methods in challenging situations. Section VII exposes common odometry limitations as well as some possible solutions, and section VIII presents the main conclusions arising from this research.

II. VISUAL ODOMETRY

Odometry is the process of estimating an agent's change in position and orientation over time. Visual odometry (VO) is the designation given when relying on the input of a single or multiple cameras attached to the agent. VO methodologies consist of reckoning the pose of the sensor (or system where it is mounted, e.g., autonomous vehicle) by extracting ego-motion parameters from correspondences between sequential image frames.

Given the agent's pose in timestep $k-1$, X_{k-1}, in a fixed frame, the goal of visual odometry is to compute the transformation T_k^{k-1} (Equation 1) , such that $X_k = T_k^{k-1} X_{k-1}$. This operation allows to retrieve an estimate of the pose in timestep k, X_k, by relating the different camera perspectives of successive frames.

$$T_k^{k-1} = \left(\begin{array}{c|c} R_k^{k-1} & t_k^{k-1} \\ \hline 0 & 1 \end{array} \right) \tag{1}$$

$R_k^{k-1} \in SO(3)$ and $t_k^{k-1} \in \mathbb{R}^3$ are the rotation and translation, respectively, between poses in time-steps $k-1$ and k. The vehicle's trajectory up to a timestep k, can thus be reconstructed by integration from the initial pose X_0, following Equation 2.

$$T_k^0 = T_1^0 T_2^1 ... T_k^{k-1} \tag{2}$$

The set of existing VO methods can be divided into two distinct groups: knowledge-based and learning-based approaches. The first exploits camera geometrical relations to assess the motion, whereas the other is based on Machine Learning techniques, which rely on considerable amounts of data to acquire pose prediction capabilities. As illustrated in Figure 1, knowledge-based methods can be categorized into three sub-groups: appearance-based, feature-based, and hybrid.

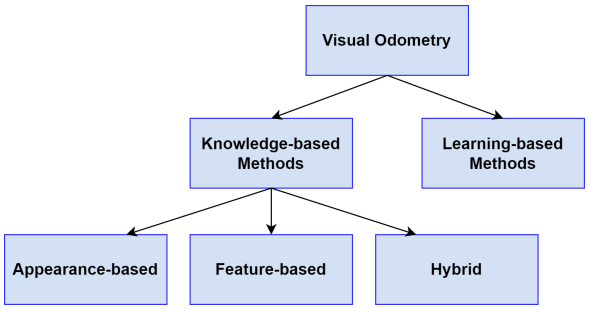


FIGURE 1: Visual Odometry categorization: the distinction is made between approaches based on Machine Learning techniques and knowledge-based ones. Inside knowledge-based techniques, the division can be made according to how visual components are used to generate odometry estimates.

A. KNOWLEDGE-BASED TECHNIQUES

Feature-based methods focus on the premise that prominent points or regions in each frame can be used to determine camera movement. These key points consist of corners, edges, lines, and blobs, which are image patterns that are distinguishable from their surroundings in terms of intensity, color, or texture, and therefore are more likely to match well across multiple images [2], [4], [5]. For feature detection, SIFT [6], SURF [7], ORB [8] and BRISK [9] are commonly employed. Feature-based VO is considerably robust to both geometric distortions, and illumination inconsistencies [10]. However, by selecting only some points of the image, some valuable information is discarded because these methods are highly dependent on correct correspondences, and therefore the presence of outliers must be minimized. Figure 2 depicts the pipeline used by feature-based algorithms as these generally follow a structured pipeline, which involves a feature detection and matching stage(or feature tracking), followed by motion estimation, and lastly, an optimization step.

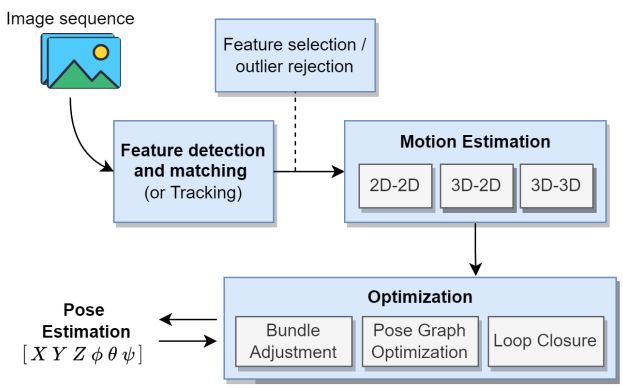


FIGURE 2: Pipeline of feature-based techniques. Although these steps are usually transversal, every technique differs in the way they propose them. The final pose estimation is composed of the agent's position in space (X, Y, Z) and orientation (roll-ϕ, pitch-θ, yaw-ψ) and can either relate to the previous pose or to a fixed global frame.

In motion estimation, the most common approaches include feature-to-feature matching (2D-2D), which exploits the constraints imposed by epipolar geometry. The epipolar constraint relates the same feature seen from different perspectives by Equation 3, where P_1 and P_2 are the image coordinates of the same point in images 1 and 2 respectively, and E is the Essential matrix. By selecting a set of matches, it is possible to calculate the motion parameters (implicit in E) that minimize the error imposed by the epipolar constraints.

$$P_1^T E P_2 = 0 \tag{3}$$

Alternatively, 3D-2D techniques resort to minimizing reprojection errors from 3D tracked landmarks against the current image frame. As the deviation from the ground-truth trajectory tends to accumulate, landmark tracking can later be useful for the optimization step as the coordinates of those points should be consistent within each camera pose, which is the principle applied in Bundle Adjustment [11]. This optimization operation is given by:

$$argmin_{X^i,C_k} \sum_{i,k} ||p_k^i - g(X^i, C_k)||^2, \tag{4}$$

where p_k^i is the i-th image point seen in the k-th image, corresponding to landmark X^i. $g(X^i, C_k)$ is the image reprojection of X^i in the camera pose C_k. From Equation 4 it can be seen that both the camera's pose and the landmarks' positions themselves are optimized. Lastly, 3D-3D techniques compare two sets of tri-dimensional points directly but are usually less accurate than the remaining alternatives.

ORB-SLAM2 [12] is a popular algorithm among the VO and Simultaneous Localization And Mapping (SLAM) communities. It is the extension of ORB-SLAM [13] for monocular, stereo, and RGB-D cameras. This open-source method is often considered a benchmark in visual odometry. By being a SLAM technique, this algorithm is composed of 3 threads: tracking and local mapping, which compose the odometry module, and loop-closing. Motion estimation is computed using ORB features tracked over keyframes and a local map, with particular emphasis on multi-step Bundle Adjustment. VISO2 [14] is another popular algorithm from 2011 that can construct 3D maps of the environment using stereo cameras. This method applies the well-known stereo matching method to match a sparse set of features in conjunction with an odometry method that uses a Kalman filter [15].

Moreover, the works of Cvišić and Petrović [16]–[18] are of special relevance in the field of visual feature-based approaches, since their results bring visual methods' precision closer to the precision levels of LiDAR-based techniques. In [16], the authors proposed a Stereo Odometry algorithm whose main focus is the careful selection and tracking of features (SOFT) and the impact of these steps on egomotion estimations. Rotation and translation are computed separately to boost the overall system performance. For rotation, Nister's 5 Points [19] is used in monocular fashion to probabilistically reduce the detrimental effect of outliers

and imperfect stereo rig calibration, while utilizing rotation to later compute translation using 3 points. Additionally, an extension of the algorithm is proposed to integrate an IMU, which estimates rotation ensemble with a Kalman filter. This first estimate helps to further reject outliers and lighten the computational cost of 5 Points algorithm, using P3P [20] and Ransac [21] instead. When the rotation is calculated via visual odometry, the Kalman state is updated. Later, Cvišić and Petrović extended SOFT with an additional mapping thread, giving rise to SOFT-SLAM [17]. This method was originally intended for autonomous unmanned aerial vehicles, focusing on computational efficiency. SOFT-SLAM integrates the SOFT visual odometry pipeline and completes it with a mapping module which adds SLAM features, such as loop-closure and global consistency constraints. These added capabilities give this technique superior localization accuracy in KITTI dataset [22] over its state-of-the-art alternatives, such as ORB-SLAM2 and LSD-SLAM [23]. Recently, Cvišić *et al.* [18] revisited the calibration parameters of the KITTI Odometry dataset. In this last work, the authors propose a new one-shot technique for calibrating the parameters of the multi-camera KITTI setup, which in turn results in smaller reprojection errors, directly impacting the accuracy of VO algorithms. The adjusted parameters were applied to ORB-SLAM2, SOFT, and VISO2, with improvements in the order of 28% in translation error and 46% in rotation error, on average.

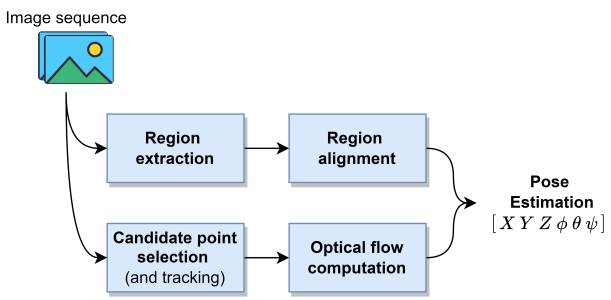


FIGURE 3: Pipeline of appearance-based techniques: the first stream refers to region matching algorithms and the second to optical-flow based techniques.

On the other hand, feature-based techniques discard a significant part of the image data by concentrating only on a few selected points. In addition, these techniques can entail extra computational costs in matching or tracking operations and outlier removal. In turn, appearance-based VO techniques make use of all the information in the captured frames instead of just using key points. These methods estimate the camera pose by analyzing the intensity of image pixels and minimizing the photometric error, which relies on the consistency principle that pixels from one frame maintain their intensity in the second frame while considering a moving sensor [5]. This way, it is possible to mitigate the aliasing effect caused by repetitive patterns and ensure more robustness in scenes with limited texture where it is harder to detect good features (e.g., foggy or sandy environments). Appearance-based techniques avoid the time required to extract and match features and run outlier rejection algorithms. On the other hand, they are sensitive to illumination variations because of the photometric principle, and abrupt camera movements compared to feature-based VO.

Appearance-based methodologies, also called direct methods, are generally classified into region matching-based or optical flow-based. The former estimates camera motion by aligning certain corresponding regions in consecutive frames, but it loses effectiveness significantly in the presence of dynamic objects in the scene, besides being susceptible to local minima solutions. On the other hand, optical-flow-based techniques resort to the optical flow of the surrounding scene to estimate the 6 Degrees-of-Freedom (DoF) of camera motion based on motion models. Figure 3 aggregates both approaches in the generalized pipeline of direct techniques. In an effort to mitigate the disadvantages of both feature-based and appearance-based methods, as well as to aggregate the added value of each, it is also possible to resort to different approaches from each domain. These are called hybrid techniques [5].

In 2017, Engel *et al.* [24] proposed Direct Sparse Odometry (DSO). This method comprises a direct and sparse scheme, thus not requiring feature detection and matching operations. This approach works by continuously optimizing the photometric error over a finite window of frames. However, in contrast to typical direct methods, the optimization occurs for all parameters simultaneously, including egomotion, camera calibration, and inverse depth of 3D points. By considering the full error, as opposed to the error of a particular iteration only, this strategy limits the effect of outliers. The authors conclude that with proper hardware (global shutters, precise lenses, and high frame rates), direct formulations could surpass geometrical/indirect approaches in terms of accuracy, which have dominated research interest in the past decade.

Traditional (or geometric) methods already have well-established foundations, and while these approaches have generated reasonable results throughout their evolution, they still prove to be fragile in environments with increased complexity. In fact, it becomes challenging to rely solely on these types of techniques since it is extremely difficult to capture the complexity of the real world by hand formulation.

B. LEARNING-BASED TECHNIQUES

As seen in diverse areas, data-driven learning-based approaches can acquire a high-level understanding of the scene without the need for explicit modeling, as long as they are trained on sufficiently large-scale representative datasets [25]. In addition, camera calibration parameters do not need to be known *a priori*; translation can be estimated with the correct scale, and the system becomes more robust against image noise [26]. As a result, there has been a paradigm shift in VO, leaning towards learning-based methods in recent years. Typical data-driven techniques often consist of multi-

ple sub-nets with distinct functions, such as depth estimation, feature extraction, and ego-motion estimation [25]. These can be either used to supplement traditional pipelines or build an end-to-end architecture, as depicted in Figure 4. The networks are trained by confronting the outputs with a supervision signal or evaluating them against a cost function that readjusts and fine-tunes the network parameters.

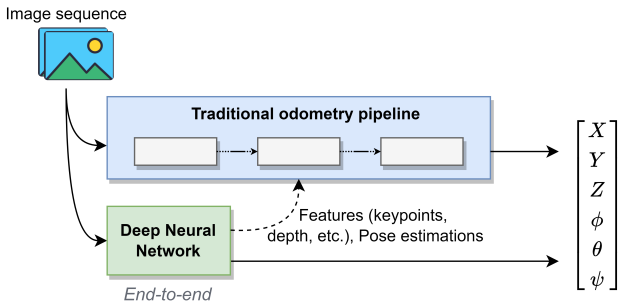


FIGURE 4: Pipeline of learning-based techniques. Deep learning networks can either complement traditional odometry pipelines, or directly output pose estimates.

Yang *et al.* [27] proposed Deep Virtual Stereo Odometry (DVSO), which complements DSO with a Deep Learning-based framework. The core of this work is based on extending the capabilities of DSO through a network that produces accurate depth estimates, thus reducing scale drift. StackNet, the proposed Fully Convolutional Network (FCN), generates a pair of depth maps that simulate a virtual stereo rig. Training is achieved in a self-supervised manner by comparing the back-warped produced outputs with the original inputs. This technique is common in odometry learning-based methods, so to avoid the use of ground truths, which are costly to obtain. StackNet's predicted depth maps are added as additional geometric constraints to the estimated depths in the original DSO general optimization problem. Deep visual odometry methods have so far failed to supplant classical ones. This work, however, goes beyond the learning-based alternatives and comes closer to conventional state-of-the-art techniques. In fact, DVSO obtained a slight improvement of 2.2% in translation error over ORB-SLAM2 in a set of KITTI selected sequences. Moreover, although DVSO is not an end-to-end visual odometry architecture, as DeepVO [28], DVSO translation error is 9.3 times smaller in the KITTI dataset.

In 2020, Yang *et al.* [29], the same authors of DVSO, presented Deep Depth, Deep Pose and Deep Uncertainty, also known as D3VO. The overall architecture of D3VO is slightly different from that of DVSO and DSO. In D3VO, depth is estimated from a convolutional network called DepthNet, which also predicts the uncertainty associated with the estimates. Camera poses, in turn, are estimated from another deep convolutional network called PoseNet, which, in addition to transformations, can equalize the illumination of the current and previous frames that together constitute the pair of network inputs to mitigate errors arising from the variation of lighting. Uncertainty is particularly important in this equalization process as it helps detect non-Lambertian reflective surfaces that easily violate the photometric consistency principle between frames. Both DepthNet and PoseNet are trained in a self-supervised and joint manner. The pose predictions complement a direct tracking front-end module and a global Bundle Adjustment back-end module based on previous works DVSO and DSO. The results slightly exceed those of DVSO (10% in selected KITTI sequences) regarding trajectory accuracy.

Unlike the previous approaches, the work presented by Wang *et al.* [28], DeepVO, is an end-to-end supervised method whose focus goes on learning feature extraction with proper geometric significance and implicitly modeling the motion dynamics over a sequence of frames. A pair of sequential images are stacked and passed as an input to a Convolutional Neural Network (CNN), generating a compact descriptor of the input pair, which is then fed to a Recurrent Neural Network (RNN), allowing to capture the sequential properties of VO. Nonetheless, unlike DVSO and D3VO, the results are somewhat unsatisfactory. Yet, this approach allows for a non-tuning-dependent model that works as a proof of concept for end-to-end DL methodologies. In fact, most of the commonly used DL structures in computer vision, such as CNNs and RNNs, are not well suited for VO. This kind of work can serve as a starting point and catalyze new VO applied research. ESP-VO [30] extends this work by calculating uncertainties of pose estimates, which is particularly useful for sensor fusion. Following a similar path, PoseConvGRU [31] is an end-to-end comparable method. PoseConvGRU leverages DeepVO time efficiency by using stacked Gated Recurrent Units (GRUs) instead of Long Short-Term Memory modules (LSTM). LSTMs and GRUs are specific types of RNNs that capture long-term relations between successive inputs by learning and storing internally what is relevant to keep, or not, in the form of internal states. This type of structure is especially important in visual odometry, given the temporal geometrical constraints over a sequence of frames. In PoseConvGRU, GRUs are preferred since they are quite close to LSTMs in performance terms but achieve similar results with fewer parameters and less time consumed. Nevertheless, while achieving slightly better results than DeepVO, the improvement is not significant, thus leading to the same conclusions

DeepAVO [32] is another DL approach based on optical flow, which relies on a learning-based optical flow extractor, PWC-Net [33]. It presents a four-branch network for each image quadrant to exploit local visual cues. A Convolutional Block Attention Module (CBAM) [34] mechanism is incorporated into the feature encoder prior to pose estimation. This mechanism acts as a mask to distill relevant features, focusing on pixels in distinct motion and discarding foreground and blurred objects. Also relying on PWC-Net, Zhao *et al.* [35] predict the optical flow to compute the relative transformation of the camera pose and reconstruct a few 3D structures of the scene by triangulation. These structures are then used to align depth predictions coming

from a parallel neural network, tackling the problem of scale inconsistency between pose and depth predictions. This is a problem that hinders the learning process and, therefore, the results, so the authors divide the training process of pose and depth branches, taking advantage of the output of both to complement each other and create consistency between them.

ClusterVO [36] stands out as a dynamics-aware VO technique that can segment dynamic objects while retrieving the trajectory of the camera and the trajectory of the detected objects. The motion estimation part is based on using keyframes and sliding window optimization (partly similar to ORB-SLAM2). VLocNet [37] presents an approach based on Auxiliary Learning applied to visual odometry. In addition to a VO dedicated network, the authors propose another module to estimate the agent's global pose, sharing features between the two since both tasks are very similar in nature. This practice promotes a more consistent learning and less susceptibility to overfitting. This technique was further extended in VLocNet++ [38] by including a scene segmentation task.

So far, the visual methods landscape has been dominated by traditional feature-based methods, like ORB-SLAM2 and SOFT-SLAM, for example. More recently, direct formulation methods have been achieving interesting results, while we are beginning to observe a race to integrate learning-based submodules to complement the more traditional architectures, as in the case of DVSO and D3VO. In parallel, and still beyond the classical alternatives, many end-to-end approaches for ego-motion estimation have emerged, which despite the underwhelming results, present great potential in terms of abstraction to the complexity of road traffic environments.

III. POINT CLOUD-BASED ODOMETRY

Maintaining the problem setting already established in section II regarding VO, where the objective revolves around obtaining the translation, t_k^{k-1}, and rotation, R_k^{k-1}, between poses of time steps $k-1$ and k, some changes are necessary to completely establish point cloud-based approaches. Frames are typically represented in the form of point clouds, which compose a set of 3D points given directly in world coordinates, contrary to VO, where points are given in pixel/image coordinates. LiDAR sensors work by measuring the time of flight or phase shift of an emitted and reflected laser ray. To create a 3D point cloud of the environment, it is necessary to perform this operation multiple times to cover the entire scanning area.

The categorization of point cloud-based odometry methods is not as straightforward as in the case of VO. The most simple way of dividing these types of work involves categorizing them into one of the following: knowledge-based, where standard and conventional algorithms are used, and learning based that involve the use of machine and Deep Learning techniques.

The following sections contain a summarized description of a general LiDAR-based odometry pipeline as well as brief descriptions of the methods that constitute the current state of the art.

Jonnavithula *et al.* [39] help define the basic steps in laser odometry algorithms: (1) pre-processing, (2) feature extraction, (3) correspondence searching, (4) transformation estimation, and (5) post-processing (Figure 5).

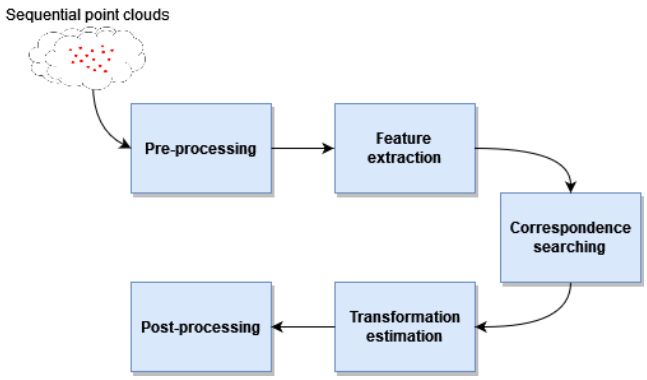


FIGURE 5: General LiDAR-based odometry pipeline.

In pre-processing, point clouds are restructured into more convenient forms, usually by segmentation operations or 2D projections, such as a range images. The last are especially useful when working with CNNs, for example. Some procedures like ground and dynamic object removal are also applied in this step. It is also important to note that the LiDAR collected point cloud is typically unstructured, i.e., it consists of a set of points in which the concept of neighborhood is not explicit, as it is in adjacent pixels. In other words, there is no information about vertex connection or about the surface underneath either. By searching the vicinity of each data point, it is possible to retrieve that kind of information and store it in a multitude of data structures, like meshes, voxels, octrees, and K-D trees, which speed up the searching process. Feature extraction can be done much like in VO, using extractors like SURF [7] and SIFT [6] if the point cloud is transformed into a 2D representation, as mentioned before. 3D features can also be extracted from point clouds using the concept of local features that encode the shape of small patches around a set of specific key points. This step can help avoid storing many points while facilitating the matching process. In correspondence searching, the objective is to find point correspondences between successive frames. This could be done in a variety of ways, like in Iterative Closest Point (ICP) [40] where all points are considered, or by using unique feature descriptors to create correspondences. In transformation estimation, once point correspondences are known, it is possible to calculate the transformation matrix T_k^{k-1} for each pair of correspondences. The general pose transformation between successive frames usually minimizes the displacement of all matching points in the point clouds. Post-processing varies from method to method, but it usually refers to some iterative refinement process as loop closure. Loop closure occurs once a point in the map is crossed more than once, adding extra pose constraints to the algorithm. This operation, however, requires a dedicated mapping and optimization mechanism.

A. KNOWLEDGE-BASED TECHNIQUES

ICP [41] had its first versions in the nineties and can be considered one of the most influential algorithms in the area of LiDAR-based odometry. In its most basic version, this method finds the transformation between two point clouds in two steps: data association and transformation estimation. The data association step aims to find correspondent points between two point clouds, which can be done using a Nearest Neighbor approach. The second step aims to minimize the distance between point pairs by first computing the center of mass of each point cloud and aligning them, then computing the rotation using Single Value Decomposition (SVD). This algorithm is run multiple iterations until a local minimum is found. The basic concepts of ICP are used in a variety of state-of-the-art approaches, for example, CT-ICP [42], which is one of the best performing odometry systems on the KITTI autonomous driving benchmark. CT-ICP adapts the ICP algorithm to work in real-time, taking into consideration the distortion of point clouds caused by sensor motion, as it happens in the autonomous driving scenario. This formulation makes this approach robust to high frequency movements of the sensor, which is the method's main strength. The algorithm estimates initial and final positions for each LiDAR scan while performing an elastic scan matching through interpolation. Unlike similar methods, the final pose does not necessarily correspond to the initial pose of the next swipe, providing elasticity and robustness to more abrupt variations in sensor movement. In parallel, this method provides a complete mapping module with a novel loop closure procedure. The map's points are inserted into a 2D elevation grid by clipping the z coordinate between certain thresholds, which only works if the sensor's movement is relatively stable on the z-axis. When a new grid is built, it is matched against older ones, using rotation invariant 2D features. When a match is validated, ICP is used to refine the 2D transform obtaining a 6-DoF loop closure constraint.

Another recent approach that relies on the basics of ICP is MULLS (Multi-metric Linear Least Square) [43] which provides an efficient, low drift 3D SLAM system. This architecture is especially designed to be independent of the LiDAR's specifications, not needing the conversion of the LiDAR data to rings or range images. Firstly geometric feature points are extracted and encoded, distinguishing between several categories such as ground, facade or pillars. The next step involves the ego-motion estimation by multi-metric linear least square ICP, based on the selected features, which is modified to increase the accuracy and efficiency. This variation has four essential steps. First, point correspondences are determined within each feature category. After that, weights are calculated for each correspondence considering several factors such as point intensity. Then, according to the point correspondences and calculated weights, the transformation estimation is computed. Finally the authors use statistical metrics to evaluate the quality of the registration procedure. An example of the MULLS registration procedure performed on LiDAR collected point clouds is depicted in Figure 6.

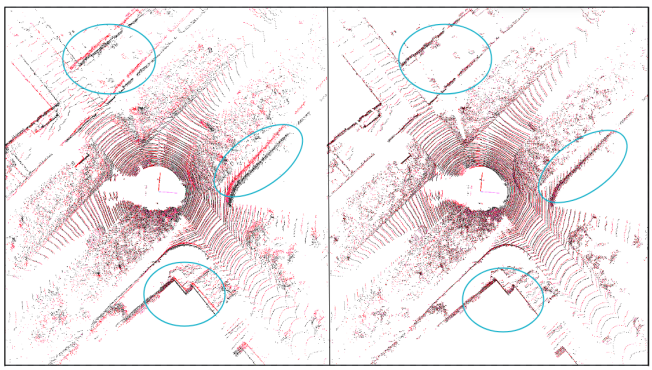


FIGURE 6: Visualization of the MULLS registration procedure highlighted within the circles. Different point clouds are represented with different colors: red and black . Left image: before registration, right image: after registration.

Another work that is of high relevancy in this area is LOAM [44]. This method addresses both odometry and surrounding mapping at different frequencies. The odometry function operates at a higher cadence and generates low fidelity estimates. At the same time, the mapping module is slower and adjusts the odometry estimates while generating and refining a global map. Regarding odometry, point cloud features are extracted in a preliminary stage, selecting and grouping points into sharp edges and planar surface patches. Along the LiDAR sweep, the captured points are progressively projected onto the previous frame, assuming the sensor's constant angular and linear velocities. As features are extracted and matched, new estimates are produced by a variant of ICP. When the sweep is completed, the mapping module refines the alignment, completing the map and producing a pose estimate with increased accuracy, passed on back to the odometry module again.

Other works, such as LeGOLOAM [45] and ELO [46], employ variations of LOAM specifically designed to have lower running times while maintaining or improving its original performance. LeGOLOAM, that stands for Lightweight and Ground Optimized LOAM, was developed to be implemented in vehicles with lower computing power and no suspensions (this aggravates point cloud distortion). The basic working principle relies on segmenting the point clouds, removing small clusters and preserving points that may represent big objects like tree trunks and road surfaces, and saving them in a range image. Then, features are extracted from these range images and classified as ground or non-ground features. Regarding the odometry algorithm itself, the matching of features across frames becomes faster because of the ground optimization procedure. As expected, this method improves the efficiency of LOAM, achieving a 72% reduction in features used. ELO [46] was recently proposed by Zheng *et al.*. It is common for methods like LOAM, for example, to encode point clouds in a tree-based form, which, while effective in terms of searching, becomes

somewhat limited in large-scale point clouds. In order to maximize efficiency, this method proposes a projection of LiDAR measurements onto a spherical image, directly restoring the neighborhoods between points. The problem arises in the fact that in autonomous vehicles, LiDAR sensors capture a significant amount of ground points, which, by the spherical geometry, become too far away in projections. As a way to take advantage of these points, the ground points are projected in a top-down bird's-eye-view perspective. With the ground and non-ground points properly segmented and with the proper 2D projections, the application of the search methods for frame matching becomes much more efficient. Concerning run-time, the authors suggest that ELO achieves 169 frames per second on a commodity laptop. In fact, this work registers a running time 21 times lower than the average for the ten top positions in the KITTI Odometry benchmark. Despite the efficiency efforts, ELO performs comparably to other methods like MULLS, for example.

Some other works should also be mentioned. F-LOAM [47], for instance, tries to reduce the computational burden by transforming LOAM's iterative processes into a two-stage distortion compensation method. It also uses special features such as edges with higher local smoothness and planar features with lower smoothness, which are good for matching. With such efforts, this method achieves a 20Hz cycle frequency on a low-power embedded computing unit. ISC-LOAM [48] is similar to F-LOAM but uses the intensity values of point clouds and their geometry to improve the capabilities in loop closure further. R-LOAM [49] takes this improvement differently by combining the LOAM framework with prior knowledge about a reference object. It requires previous knowledge of a 3D model of an obstacle and its position on a global coordinate system. However, this formulation makes R-LOAM unsuitable for the autonomous driving context.

B. LEARNING-BASED TECHNIQUES

A different type of approach involves the use of Deep Learning techniques to resolve the odometry problem. Because data from range sensors is unordered and sparse, it is challenging to apply typical convolution-based DL modules. Therefore some methods that employ DL techniques transform the 3D point clouds into other formats such as range images. LO-Net [50] is an algorithm that retrieves the odometry estimation of the sensor using this type of approach. LO-Net starts by passing from point cloud format to matrix, projecting the points into cylindrical coordinates. Then, for each point, its respective normal vector is calculated. Both the matrix containing the normal vectors from the current instant and the one from the previous LiDAR scan are passed to a Siamese Neural Network (SNN), which outputs are combined and fed to a convolutional network that estimates the ego-motion parameters. The whole network is trained in a supervised manner. Furthermore, this algorithm contains a mask estimation module for dynamic objects in the scene and a mapping block to further refine the estimates through scan-to-map matching. DeepLO [51] is another method that begins by projecting the incoming point clouds into the 2D space. The projection representation is substituted by a pair of 2D maps containing the point cloud vertices and normals without precision loss. The vertices' map from the current instant and the previous one are passed to a fully convolutional network, from which it is extracted a feature vector of the respective pair. The same applies to the normal's maps, which are fed to another similar network in parallel. The two resultant feature vectors are then summed and passed to a third neural network which then predicts the motion parameters. This network can be trained either, in a supervised manner, assisted by the sequence ground truth, or in an unsupervised manner via an error function that incorporates a version of ICP. LodoNet [52] is also worthy of mention as it transforms the 3D data into a two-dimensional representation and performs feature extraction using SIFT, obtaining keypoint pairs between successive scans. These correspondences are then fed into a convolutional neural network pipeline that extracts Matched Keypoint Pairs (MKPs). The MKPs can be accurately returned to the 3D space and fed into a convolutional neural network designed for LiDAR odometry. Another work that makes use of neural networks is PWCLO-net [53]. This algorithm learns LiDAR odometry from raw 3D point clouds in an end-to-end fashion with no need to project the point cloud into 2D representations. The inputs of the network are two point clouds, which are encoded by a siamese feature pyramid that extracts the hierarchical features of each point cloud. Then, an attentive cost volume is used to associate the two point clouds and generate point embedding features that contain point correlation information. An embedding mask is used to obtain the pose transformation from these features while also removing dynamic elements.

Other recent and relevant works that employ Deep Learning are PSF-LO [54], which uses parameterized semantic features to facilitate the registration task and employs a dynamic and static object classifier; and CAE-LO [55], that, like previous methods, uses unsupervised Deep Learning and utilizes compact 2D spherical ring projections. DMLO [56] is also an interesting work, making feature matching applicable to LiDAR odometry by decomposing the pose estimation in two parts: a matching network that makes correspondences between two scans and a rigid transformation estimation operation. SuMa++ [57] expands the previous work by Behley *et al.* [58] and has a different approach towards the self localization problem. The robot position is estimated by analyzing changes in a semantic surfel-based map, to also detect and remove dynamic objects, and improve the pose estimation. Each scanned LiDAR frame is converted into a 2D projection. Then, each frame is segmented by RangeNet++ [59], a point cloud segmentation network, as each point is attributed a semantic label. After this step, the image is converted back into a 3D projection which updates the world map.

This section provided an overview of the general concepts of point cloud-based odometry. It also analyses several works

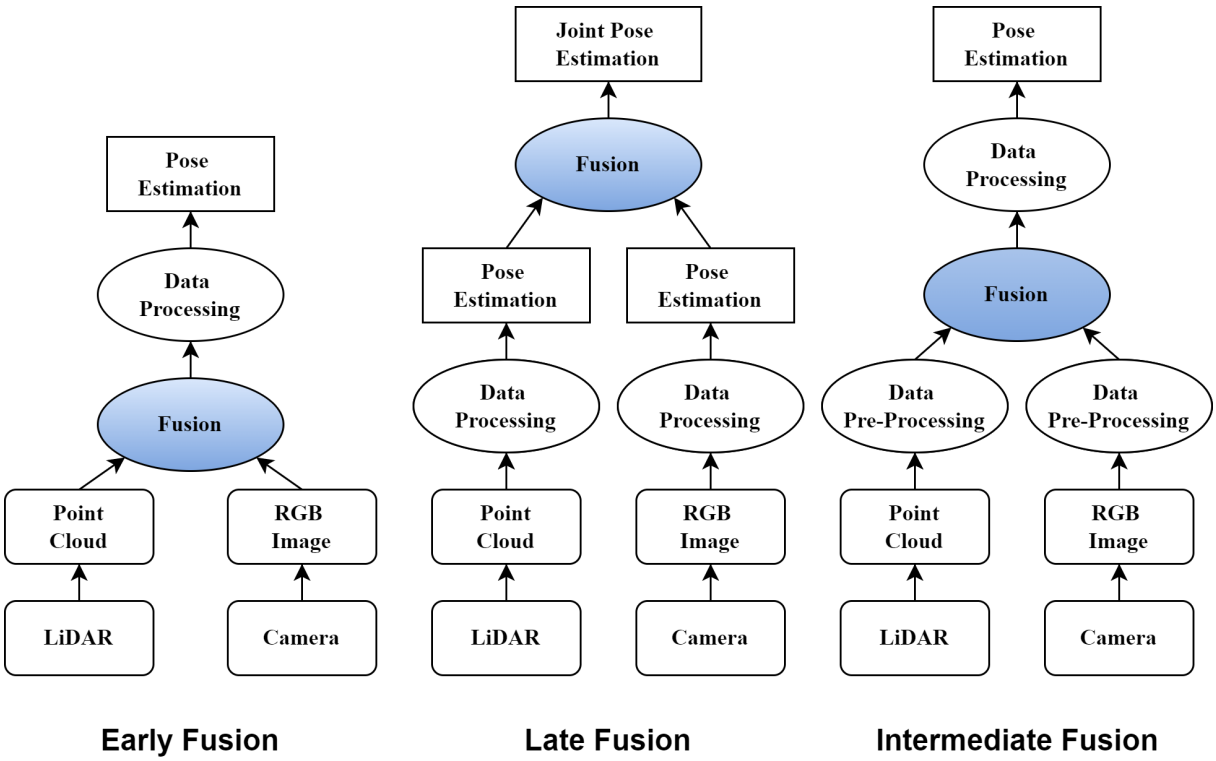


FIGURE 7: Process illustration of each data fusion technique: early, late and intermediate.

in this area, along with their unique characteristics, ranging from knowledge to learning-based approaches that are becoming more popular, just like in visual odometry. Table 1 summarizes all the referred works, while the analysis of point cloud odometry is extended in sections V and VI.

IV. MULTI-MODAL ODOMETRY

Road environments in which autonomous cars operate are highly complex. Autonomous agents can benefit from multi-modal perception by combining different sensors, such as cameras, LiDARs and IMUs, approximating the human perception capabilities. This approach is becoming increasingly relevant as these sensors start to be widely adopted by car manufacturers.

A. DATA FUSION STRATEGIES

Data fusion relies on the basis that collecting data from distinct sensors will allow capturing richer and redundant information from surroundings, which will, in turn, enable lower uncertainty in state estimation. This approach also contributes to the system's robustness by reducing the failure cost of individual sensing modalities. For example, visual odometry may suffer from hard illumination variations, poor lighting conditions, and texture-less environments; LiDAR struggles with wide-open spaces, such as motorways or long tunnels, and adverse weather conditions; and IMU's trajectory estimates tend to drift very quickly if not adjusted periodically. Thus, having more than one modality should compensate for eventual short or long-term failures.

In multi-modal approaches, data fusion can be classified based on the instant the merge takes place within the context of the system framework (Figure 7). In early fusion, data is merged in the raw stage, before any pre-processing, usually through projections of one or more sensors to the input space of another. Typically this approach is associated with low computational costs but is highly dependent on inter-sensor spatial and temporal calibration [60]. Late fusion implies an after-processing merge of data. It is the most common strategy since it entails lower complexity and increased modularity. However, it incurs higher computational expenses and blocks the use of potentially valuable intermediate features of each data type [60]. One of the most significant limitations in late fusion is relying only upon pose estimates, which confers on the fusion layer an abstraction level that can be limiting in certain situations. Finally, intermediate fusion is the more comprehensive technique, as it can be deployed in many possible ways, depending on the system's architecture, especially if DL-based. As data is fused after some processing, it is also known as feature-level fusion. In the context of odometry, one can classify how data is used to produce the system's output. For example, LiDAR measurements can complement the imagery, while estimating ego-motion with visual odometry or vice versa; or the two types of odometry can operate separately and fuse at a higher abstraction level of the system's framework [61]. The systems that follow the first approach are usually denoted as tightly-coupled,

whereas the others are called loosely-coupled [62]. In these circumstances, images and point clouds are usually the primary sources of odometry estimates, whereas IMUs and GPS systems provide expedient priors and/or trajectory drift corrections.

The current technology of digital cameras allows them to have a very competitive price and reduced size and weight compared to other sensors. Cameras record high resolution images of the surrounding scenes, by extracting color and texture information, which are especially effective for detecting contours and objects, and, in the case of VO, to identify points of interest that can be easily tracked in sequential frames. However, cameras are extremely dependent on environment conditions and illumination, facing some issues in scale recovering too. In turn, LiDAR sensors can retrieve native depth information and typically offer a wider spatial coverage. The number of channels commonly ranges from 16 to 128, and refresh rates may vary in a range from 5 to 20 Hz. Yet, rain, fog and snow can negatively impact LiDARs performance by up to 25% [63], so it is of great importance to take these phenomena into account. Some studies, as in [60], [64], demonstrated laser beams' wavelength to impact the adverse effects of weather conditions.

Calibration is also a critical aspect when merging data from distinct sensors, specially in tightly-coupled systems. Each sensor has intrinsic and extrinsinc calibration parameters that capture the internal geometrical properties of the sensors and relate the world frame with the frame of the device. Besides that, sensors must be jointly calibrated, so that multiple detections of the exact same feature, detected by different sensors, are transposed to the the exact same position in the system's common frame (ideally). The most common techniques employ physical targets with well-known dimensions. These structures must have characteristics that are easy to detect and segment by each sensor modality. The features segmented by each sensor compose a set of physical constraints that allow estimating the rotation and translation between them through parameter optimization. In [65], for instance, the authors use only a simple arbitrary flat polygon for camera-LiDAR calibration, while in [66], the authors propose a rectangular block with four tapered holes and a metallic reflector to allow the extrinsic calibration of camera, LiDAR and RADAR. In [67], the authors compile a set of openly available toolkits for sensor extrinsic calibration, while also presenting some practical considerations to have when calibrating the sensors. Some additional techniques can be found in [68]–[71], mainly for camera-LiDAR fusion, and in the *kalibr* toolkit for multi-camera and camera-IMU fusion, which employs the techniques from [72]–[76].

B. STATE-OF-THE-ART TECHNIQUES

The work presented by W. Wang *et al.* [61], DV-LOAM, is an example of how to combine LiDAR and camera data at various levels for improved ego-motion detection. DV-LOAM is composed of a front-end and a back-end part. The front-end module is divided into a VO block and a LiDAR mapping module. The first receives sequential images enhanced with sparse depth measures from the LiDAR (Figure 8), in which the first motion estimates are produced by a direct patch-based method inspired on DSO [24]. The estimates are further refined by applying a sliding window optimization technique, ensuring local consistency and real-time performance. In case the latest frame is a keyframe candidate, features are extracted from the corresponding LiDAR scan and joined to a global map, which then allows for one more refinement step in a scan-to-map process. This step allows the use of the whole LiDAR coverage angle and not only the portion corresponding to the camera perspective. The back-end module is responsible for map maintenance, including loop closure and a pose-graph optimizer, to reduce accumulated drift over long traversals. In addition, DV-LOAM presents an insightful ablation study. Conclusions suggest that the visual-LiDAR odometry method applied by this work (direct and patch-based) achieves better precision than LiDAR-only odometry as it relies on images to detect edges that are not so distinguishable in point clouds. Also, it benefits from enhancing the images with direct depth measurements, which allow the method to work even when the images are blurred.

FIGURE 8: Visualization of depth enhanced images as in DV-LOAM's front-end odometry module. This Figure illustrates the point cloud projection into the respective camera frame. Different colors represent different levels of depth.

From the same authors of LOAM, V-LOAM [77] is intended to mitigate LOAM's reliance on smooth motion. The integration of the camera into V-LOAM allows visual odometry to serve as a prior to LiDAR odometry estimates and to handle rapid motion. Analogous to LOAM's architecture, V-LOAM also operates with a dual-frequency model. The visual block computes pose transforms at a higher pace, using a feature matching method, which relies on image points whose depths are either measured directly by LiDAR or obtained by triangulation. Concomitantly, when each LiDAR sweep is complete, the new point cloud is undistorted, assuming constant linear motion. The points are registered in a local map maintained by the laser odometry block, as in LOAM. The registration allows for generating an ego-motion estimation that corrects the drift affecting visual estimates. In addition, the LiDAR-camera combination supports operation over short periods of light outbreak.

Unlike V-LOAM, LIMO, proposed by J. Graeter *et al.* [78], uses LiDAR measurements to supplement the camera images with depth information in a tightly-coupled manner.

Prominent points are selected from image frames, discarding points lying on cars and pedestrians, for example, to avoid the effect of dynamic objects. This is performed through Deep Learning semantic segmentation. The LiDAR point cloud is projected onto the image, and the depth of each feature is computed through plane fitting interpolation in its neighborhood. The fusion stage also includes foreground and ground segmentation. By matching features, a pose transform estimate is generated and adjusted by a refined back-end block that applies Bundle Adjustment. In this block, particular emphasis is placed on the careful selection of keyframes and landmarks, as well as a robustification module to mitigate the presence of outliers using trimmed-least-squares.

One further approach is Tightly-Coupled Visual-Lidar SLAM (TVL-SLAM), from C. Chou *et al.* [79], in which the visual and LiDAR modules run independently until a certain point in the pipeline when the data from both is merged, thus constituting an intermediate fusion case. Regarding the odometry domain, the visual front-end generates a preliminary pose estimate alongside the visual residuals. This estimate contributes to the calculation of the laser front-end residuals. The fusion occurs in a large-scale optimization problem whose inputs are the residuals of both modules. The inter and intra-consistency between the modules is ensured in this last step by a set of constraints that include visual landmark reprojection errors, scan-to-map-registration, and cross-constraints, as both front-ends represent the same environment. In addition, this algorithm also features a technique for extrinsic calibration between sensors and a multi-step technique for rejecting moving objects. Tests on KAIST dataset [80], which is composed of challenging scenarios in crowded road environments, demonstrated the superior performance of TVL-SLAM due to its multi-modal characteristics (88% and 78% improvements in accuracy from TVL-SLAM LiDAR only and ORB-SLAM2 respectively).

Wisth *et al.* [62] recently proposed a tightly-coupled architecture for LiDAR-visual-IMU odometry. The state estimation is formulated as a large-scale pose-graph optimization problem with multi-sensor factors. The depth of visual landmarks is calculated through the projection of LiDAR points as in [78], or via stereo matching; point clouds are undistorted to the closest image timestamp to ensure temporal synchronization, using the IMU's motion priors; plane/line features are extracted similarly to [44], which reduces the number of points in 90% for efficiency purposes. Experiments showed considerable robustness against wide spaces, dark tunnels, dense foliage, and abrupt motion. Moreover, the pose-graph formulation allows the different modalities to impact the system independently, emphasizing performance consistency in case of failure of one of the sensors.

The work of Ramezani *et al.* [81] composes a different class of methods whose primary focus is the estimation of agent odometry via an Inertial Navigation System (INS). The state of the agent's motion is propagated through the use of a multi-state constraint Kalman filter. At the same time, the integration of a stereo rig in the setup allows adding extra constraints to the vehicle motion by 2D and 3D feature matching methods, limiting the drift of the IMU estimates. This sensor is an asset in scenarios where matching successive images or point clouds is more challenging, as well as for reducing the impact of dynamic objects, as demonstrated in Section VI.

This section presented several examples of how data from different sensors can be combined to achieve more accurate results and a more robust architecture. Through direct comparisons with single-modality methods, many of the mentioned techniques showed the importance of aggregating different modalities and working with data redundancy. The analysis of multi-modal methodologies is extended in sections V and VI. Table 1 aggregates all the referred odometry techniques alongside the respective categorization, type and relevant key points.

V. ODOMETRY BENCHMARK ANALYSIS

So far, some of the most relevant and innovative works in the field of odometry have been briefly described and reviewed, including visual, point cloud-based, and multi-modal methods. Hence, this section will contain a critical analysis over some of the exposed techniques, supported by the results obtained by the authors in the KITTI odometry dataset.

KITTI dataset [22] comprises a total of twenty-two sequences of visual, LiDAR , and GPS/IMU data recorded on board of a vehicle that transits in common road traffic environments, eleven of which include the respective ground truths. Furthermore, KITTI provides an evaluation tool that compares multiple algorithms and ranks them on a scoreboard. The outcomes are evaluated in terms of three metrics: *i)* t_{rel}, the average relative translation error over sequence lengths of 100m to 800m, in percentage; *ii)* r_{rel}, the rotation error in deg/100m over the trajectory; *iii)* run-time in milliseconds. Data in Tables 2 and 3 was retrieved from the author's publications, and correspond to the evaluation of the best-performing techniques in training and evaluating sequences of the KITTI odometry benchmark.

By analyzing Tables 2 and 3, it is noticeable that very little separates the top places, regardless of their modalities or technicalities. In fact, the first three positions in Table 2 obtained a t_{rel} below 0.5%, and each method corresponds to a different sensing modality, i.e. multi-sensor, visual and LiDAR-only. The remaining places are all roughly comprised in a range from 0.70% to 0.90%, except the last two, which are DL end-to-end architectures. Analogously in Table 3, the first five positions make up a very small translation error range, in the order of 0.06%, also with representatives from each modality.

TABLE 1: Overview of the presented odometry methods: categorization and relevant key-points

Type - **V**: Visual, **L**: LiDAR, **I**: IMU
Category - **FB**: Feature-based, **AB**: Appearance-based, **DL**: Deep-Learning, **KB**: Knowledge-based,
EF: Early-fusion (Tightly coupled), **IF**: Intermediate-fusion (Tightly coupled), **LF**: Late-fusion (Loosely coupled)

Method	Type	Category	Keypoints
ORB-SLAM2 (2017) [12]	V	KB(FB)	- Open-source and well documented - Multi-step bundle adjustment - Works for monocular, stereo and RGB-D cameras
SOFT/ SOFT-SLAM (2015/2018) [16], [17]	V	KB(FB)	- Careful feature selection and tracking (SOFT) - Efficient mitigation of outliers' effect
SOFT2 (2021) [18]	V	KB(FB)	- One shot technique for calibrating the camera setup parameters
DSO (2017) [24]	V	KB(AB)	- Continuous optimization of photometric error - Ego-motion, depth and camera calibration joint optimization
DVSO (2018) [27]	V	KB(AB)/ DL	- Complements DSO with depth estimates using a deep neural network - Self-supervised
D3VO (2020) [29]	V	AB/DL	- Depth, estimates' uncertainty and pose given by deep convolutional networks - Illumination equalization between frames - Self-supervised - Uncertainty helps to remove the effect of non-Lambertian surfaces
DeepVO (2017) [28]	V	DL	- End-to-end architecture with CNN+RNN - Supervised learning - Learns effective feature representation and sequential dynamics
ESP-VO (2018) [30]	V	DL	- Extends DeepVO by computing uncertainties of pose estimations (relevant for sensor fusion)
PoseConvGRU (2020) [31]	V	DL	- Employs stacked GRUs instead of LSTMs as in DeepVO for increased time and memory efficiency
DeepAVO (2022) [32]	V	DL	- Optical-flow based. Uses PWC-Net. - Exploitation of local visual cues by dividing input images in four quadrants and implementing a four-branch network - Incorporates a CBAM attention mechanism to distill relevant features
ClusterVO (2020) [36]	V	DL	- Motion estimation module is similar to ORB-SLAM2 - Segments dynamic objects while retrieving their motion
VLocNet/ VLocNet++ (2018) [37], [38]	V	DL	- Auxiliary learning applied to visual odometry - Scene segmentation (VLocNet++)
Zhao *et al.* (2020) [35]	V	DL	- Detangles depth and pose estimation networks to boost the odometry learning process - Pose and depth estimation with scale consistency
CT-ICP (2021) [42]	L	KB	- Adapts the ICP algorithm to work in real time - Robust to high-frequency movements by differentiating time during and between scans
MULLS (2021) [43]	L	KB	- Divides geometric feature points into several categories (ground, facade, ...) - Applies a variation of ICP, designed to work with several feature categories
LOAM (2014) [44]	L	KB	- Mapping and odometry algorithms which work at different frequencies - Extracts feature points located on sharp edges and planar surfaces
LeGOLOAM (2018) [45]	L	KB	- Low computing power requirements - Ground optimization procedure reduces the amount of used features, enabling higher working frequency
ELO (2021) [46]	L	KB	- Transforms the point clouds onto a spherical image for faster point search - Efficient frame matching procedure, based on ground points
R-LOAM (2021) [49]	L	KB	- Uses prior knowledge about a reference object to improve accuracy - The 3D model and position of the reference object should be known *a priori*
F-LOAM (2021) [47]	L	KB	- Uses two-stage distortion compensation instead of traditional iterative method - Low computational cost, using only edge and planar features with specific characteristics
ISC-LOAM (2020) [48]	L	KB	- Explores the intensity property from point clouds and their geometry to aid in place recognition

PWCLO-Net (2021) [53]	L	DL	- No need to pre-project the point clouds into the 2D space - Uses siamese feature pyramid encoder - Removes dynamic features
Lo-Net (2019) [50]	L	DL	- Scan to scan estimation network that learns the normal and masks dynamic objects - Couples mapping module into the estimation pipeline
LodoNet (2020) [52]	L	DL	- Convolutional neural network pipeline for LiDAR - Transforms point clouds into 2D spherical depth images to extract matched keypoint pairs (MKP) - SIFT for feature extraction - MKPs can be reprojected to the 3D space
Suma++ (2019) [57]	L	DL	- It presents localization and mapping features (SLAM) - The generated map has semantic information extracted by a fully convolutional neural network - Semantics are used to filter moving objects
DeepLO (2019) [51]	L	DL	- Projects the point cloud into the 2D space - Maps containing the point cloud vertices and normals are extracted from the projections - The maps are fed into FCNs and the resulting tensors are used to infer the movement - Trainable by supervised and unsupervised frameworks
PSF-LO (2021) [54]	L	KB/DL	- Uses four types of parameterized semantic features and geometric features to extract the odometry estimation - Dynamic and static object classifier with several parameters such as velocity and temporal heading
CAE-LO (2021) [55]	L	DL	- Converts point clouds onto spherical ring projections - Employs convolutional auto-encoder networks that detects key points in the 2D projections - Unsupervised learning
DMLO (2020) [56]	L	DL	- Decomposes pose estimation in two parts: correspondence match and rigid transformation estimation - Encodes the point cloud onto a 2D image
DV-LOAM (2021) [61]	V/L	KB EF/LF	- VO front-end uses images enhanced with depth from LiDAR in direct way inspired on DSO - Back-end module maintains a depth map and refines VO estimates, using full LiDAR coverage - Back-end module supports loop closure and pose graph optimization to reduce accumulated drift over long traversals
V-LOAM (2015) [77]	V/L	KB/LF	- Camera supporting extension of LOAM - Dual-frequency architecture - Images provide fast-pace odometry estimates which are periodically adjusted by the LiDAR registration module
LIMO (2018) [78]	V/L	KB/EF	- Depth enhanced images by point cloud projection and plane fitting interpolation - Foreground/ground segmentation - Deep segmentation of dynamic objects - Trimmed-least-squares for additional outlier rejection
TVL-SLAM (2021) [79]	V/L	KB IF/LF	- Visual and LiDAR odometry modules run independently up to a certain extent - Mid-fusion of visual and LiDAR residuals - Fusion takes place in a large scale optimization problem - Feature map and local LiDAR map - Features a method for extrinsic sensor calibration, and a technique for rejecting moving objects
Ramezani *et al.* (2018) [81]	V/I	KB/LF	- The agent's motion is propagated using an INS and a Multi-State Constraint Kalman filter (MSCKF) - The included stereo rig adds additional constraints to the state estimation via feature matching
Wisth *et al.* (2021) [62]	V/L/I	KB/EF	- State estimation formulated as a large scale pose graph optimization problem with visual, LiDAR and IMU factors - Tracking of visual and LiDAR features (based on LOAM) - Sensor fusion in multiple levels, e.g. depth enhancement of images as in LIMO and point cloud distortion with IMU

It may also be observed that conventional techniques predominate when compared to learning-based ones. In fact, during the last years, only small improvements occurred in terms of trajectory accuracy. Eventually, the focus will have change to further robustify the odometry systems, and make them more computationally efficient. Unlike the runtime or the data allocated by an algorithm, quantifying and measuring robustness is not a trivial task. The most common way to put the techniques to the test is by evaluating them in datasets containing challenging scenarios. Although KITTI is vastly used, and therefore suitable for comparisons, some other datasets such as KAIST [80] include more challenging sequences. In this regard, multi-modal systems achieve better results, as shown by the experiments of C. Chou *et al.* [79] (TVL-SLAM) in KAIST, and D. Wisth *et al.* [62], which evaluated their respective works against unfavorable conditions targeted at specific sensor modalities, such as the ones discussed in section VII, both standing up to the tests and showing no signs of overall degeneration, unlike visual or LiDAR-only methods. In terms of efficiency, ELO stands out unrivaled while also performing very competitively in terms of accuracy. On the other hand, complex methods like TVL-SLAM have an increased computational load, since they rely on multiple modules running in parallel to achieve such precision and robustness, some of which can be demanding, such as optimization and mapping modules. One possible solution for this type of architecture would be to enhance individual function blocks, e.g. integrate ELO or a similar approach into the laser odometry front-end.

TABLE 2: Results in KITTI training sequences (00-10).

	Modality	t_{rel} (%)	r_{rel} (°/100m)
TVL-SLAM [79]	V/L	0.47	NDA
SOFT2 [18]	V	0.48	0.11
ELO [46]	L	0.50	0.18
SOFT-SLAM [17]	V	0.68	0.2
ORB-SLAM2 [12]	V	0.72	0.22
V-LOAM [77]	V/L	0.75	NDA
DV-LOAM [61]	V/L	0.80	0.38
F-LOAM [47]	L	0.80	0.44
D3VO* [29]	V	0.82	NDA
LO-Net* [50]	L	0.83	0.4
LOAM [44]	L	0.84	0.5
DVSO* [27]	V	0.89	0.2
DeepVO* [28]	V	5.96	6.12
DeepLO* [51]	L	9.59	3.99

V: Visual **L**: LiDAR *: Learning-based methods

VI. BENCHMARK OF OPEN-SOURCE METHODS

In order to expand on previous approaches [82], to perform a well-structured and unbiased evaluation of odometry estimation, some open-source methods were selected. These techniques were tested against a set of challenging sequences extracted from the KITTI-360 dataset [3], which vary from a regular drive on quiet residential streets to busy motorways.

TABLE 3: Results in KITTI testing sequences (11-21).

	Modality	t_{rel} (%)	r_{rel} (°/100m)
SOFT2 [18]	V	0.53	0.09
V-LOAM [77]	V/L	0.54	0.13
LOAM [44]	L	0.55	0.13
TVL-SLAM [79]	V/L	0.56	0.15
CT-ICP [42]	L	0.59	0.14
SOFT-SLAM [17]	V	0.65	0.14
ELO [46]	L	0.68	0.21
D3VO* [29]	V	0.88	0.21
DVSO* [27]	V	0.9	0.21
LIMO [78]	V/L	0.93	0.26
ORB-SLAM2 [12]	V	1.15	0.27

V: Visual **L**: LiDAR *: Learning-based methods

The obtained results, as well as the respective analysis are presented throughout this section.

KITTI-360 is the successor to the well-known KITTI [22] autonomous driving dataset. It improves on its previous iteration by adding sensors like a pair of fisheye cameras and an additional laser scanner along with longer and more complex drives. In this way, eleven sequences were extracted and briefly described in Table 4. The first two digits of each sequence ID correspond to the drive in the KITTI-360 dataset. These sequences compose a variety of environments especially selected to challenge odometry algorithms and assess the corresponding limitations. Some challenges include high brightness variations, the presence of many dynamic objects, sensor blockage, and wide-open spaces, among others.

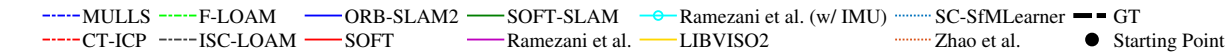
The selected evaluation scenes were used to test a few visual odometry methods in different scenarios to assimilate the state-of-the-art performance in different conditions, while discussing the limitations of each modality. Some of the most popular and best-performing algorithms were chosen:

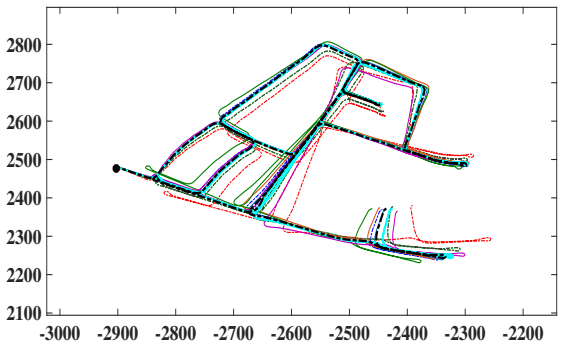
- Visual odometry: ORB-SLAM2 [12], LIBVISO2 [14] and open source implementations of SOFT[1] [16] and SOFT-SLAM[2] [17]. The deep-learning approaches SC-SfMLearner [83] and the work of Zhao *et al.* [35] were also tested.
- Point cloud-based odometry: MULLS [43], CT-ICP [42], F-LOAM [47], and ISC-LOAM [48].
- Multi-model odometry: the work of Ramezani *et al.* [81] that fuses a stereo rig with an IMU.

These methods were selected since they are open source and well-documented, allowing anybody to implement them. Some, like ORB-SLAM2, are considered to be landmarks in the area of VO, and others like CT-ICP are top performers on the KITTI odometry benchmark. This set gives a good understanding and representativeness of all visual odometry categories including knowledge and learning-based works.

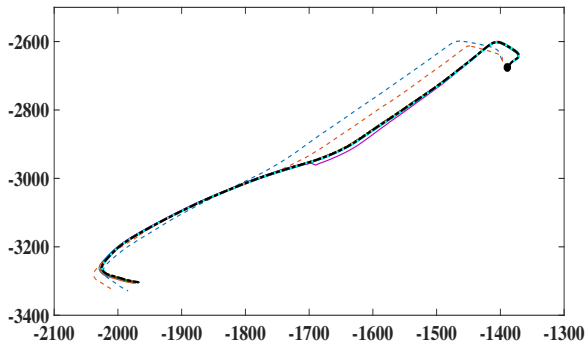
[1]https://github.com/Mayankm96/Stereo-Odometry-SOFT, 28th of march 2022

[2]https://github.com/ZhenghaoFei/visual_odom, 28th of march 2022

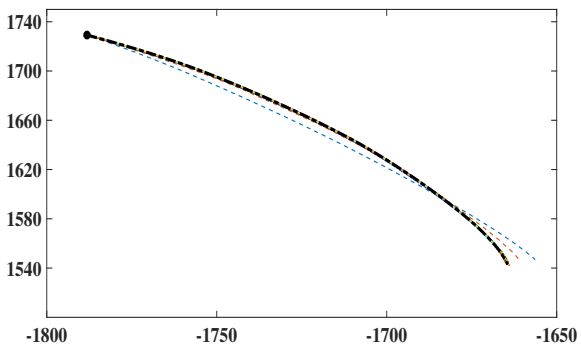




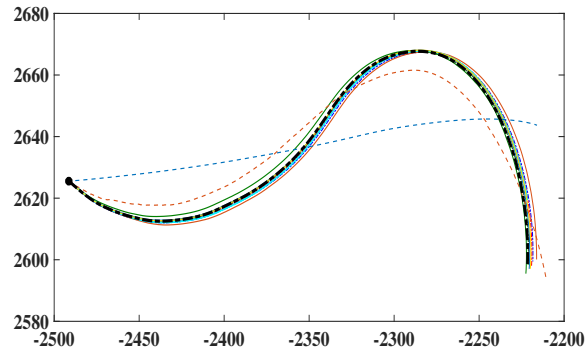
(b) Sequence 00_01



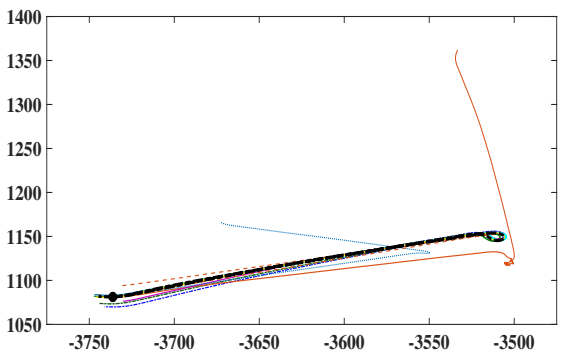
(c) Sequence 03_01



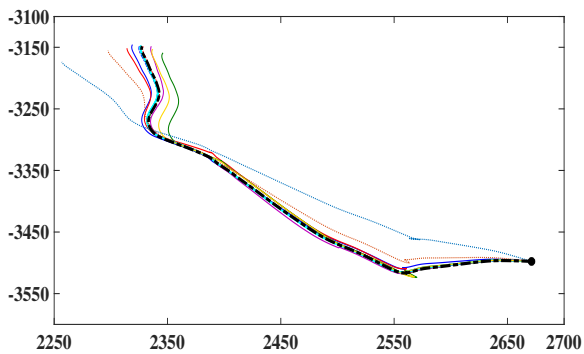
(d) Sequence 03_02



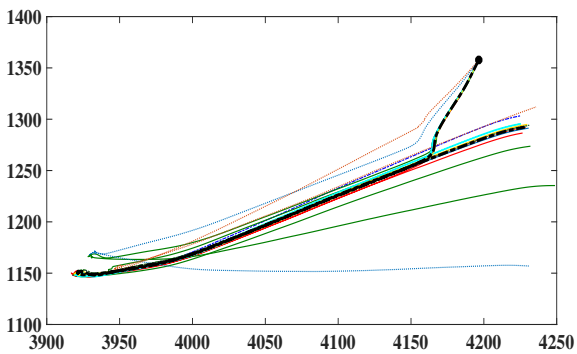
(e) Sequence 04_01



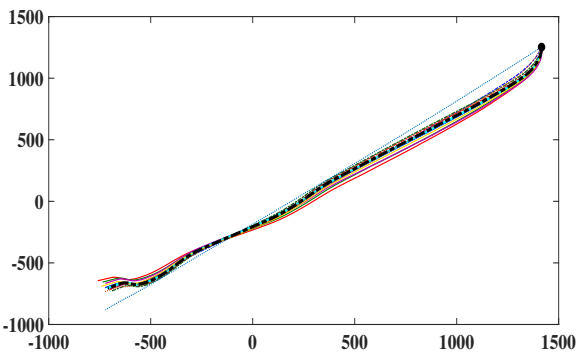
(f) Sequence 04_02



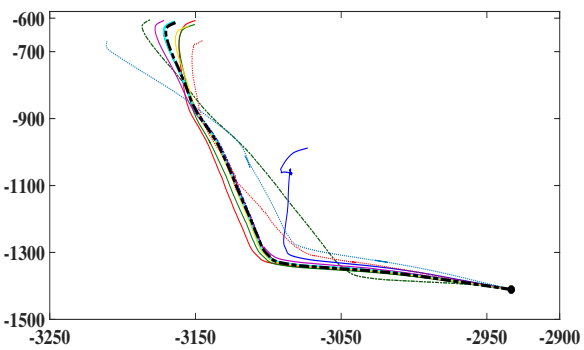
(g) Sequence 05_01



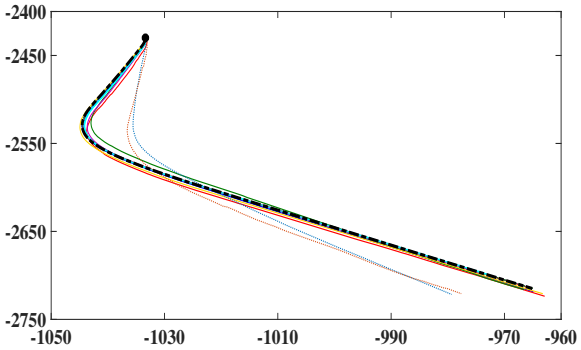
(h) Sequence 06_01



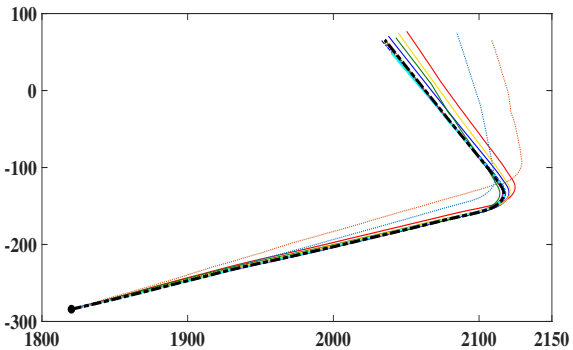
(i) Sequence 07_01



(a) Sequence 07_02



(b) Sequence 10_01



(c) Sequence 10_02

FIGURE 9: Trajectories of evaluated odometry techniques in the selected sequences of KITTI-360 dataset. Axis units in meters.

TABLE 4: Description of evaluated sequences from KITTI-360 dataset.

ID	Type (Nº frames)	Description	Long sequence	Turnaround maneuvers	Dense vegetation	Open spaces	Brightness variations	Dynamic objects	High velocity
00_01	Suburbs (5001)	Long scene, with several turnaround maneuvers	++	+					
03_01	Country (852)	Wide country road surrounded by dense vegetation			+		+		
03_02	Country (141)	Country road surrounded by dense vegetation			++		+		
04_01	Country (293)	Wide country road surrounded by dense vegetation			+		+		
04_02	Suburbs (527)	Back and forward along a residential road		+					
05_01	Road (863)	Very busy road, with heavy trucks that cause sensor blockage		+				+++	
06_01	Country (1004)	Road surrounded by dense vegetation			+				
07_01	Country (1426)	Fast moving road, dense vegetation as well as wide open fields	+			+		+	+
07_02	Highway (1090)	Highway entrance scenario with a lot of moving vehicles						++	+
10_01	Urban (227)	Wide urban roads, busy traffic and a tunnel passage					++	+	
10_02	Urban (434)	Urban roads with a tunnel passage					++		

A. ANALYSIS ON CHALLENGING SEQUENCES

Two evaluation metrics were used to test the performance of each method. Considering that the predicted trajectory and ground truth consist of a set of points, the first evaluation metric is calculated by averaging the absolute translation error between the predicted pose and the ground truth for each set ($\overline{t_e}$). The second metric, t_{seq}, consists of dividing each test sequence into sub-sequences of 100m, 200m, 300m, and so forth up to 800m and calculating the average translation error for each stretch, later condensed into a global average for all sub-sequences. The gathered results can be consulted in Table 5, where the best results are underlined, as well as in Figure 9, which comprises the trajectories described by each method in each sequence. The performance of the evaluated methods will be discussed in the following paragraphs, specifically concerning some particular challenges.

(a) Road with dense vegetation and lighting variations.

(b) Highway scenario.

FIGURE 10: A few challenging scenes from sequences 03_02 and 07_01. Dense vegetation and open spaces can degrade the accuracy of pose estimates [3].

1) Vegetation and wide open roads

In sequence *03_02*, the vehicle drives through a short straight road with dense vegetation surrounding the totality of the road margins, in all its extent, as shown in Figure 10a. Vegetation is a particular challenge for camera and LiDAR techniques since the repetitive and noisy patterns make feature matching/tracking difficult for visual methods. The dense foliage promotes defective reflection of laser beams due to the irregularity of the surfaces and multiple small occlusions, which hinders the precise alignment of point clouds. Despite generally better results, LiDAR-based methods are outperformed by purely visual ones, with a 55% lower average error. The reason is that, possibly, visual techniques rely on visual cues on the road that LiDARs do not detect. Besides that, the cameras' vertical field of view is slightly wider than the LiDAR's, benefiting from more distinct features since the vegetation is not as dense on top.

Sequence *07_01* contains relatively wide open roads (Figure 10b), which constitute challenging cases for LiDAR techniques since the open spaces make the alignment of point clouds difficult due to the scarcity of features. In this sequence, the best-performing visual method (ORB-SLAM2) achieves a 20.3% lower translation error than CT-ICP, which is the best-performing LiDAR-based technique. It is also worth noting the immunity of the IMU, as it is a proprioceptive sensor, to external factors in the scene, as Ramezani *et al.* (with IMU) is second-best in *03_02* and best in *07_01* by a large margin.

2) Turnaround maneuver and tunnel passage

As depicted in the trajectory plot of Figure 9h, sequence *06_01* has a turnaround maneuver in which the 360-degrees spatial coverage of point cloud-based approaches helps for a more precise rotation estimation. In the Figure, it is possible to witness the most pronounced deviations occur for visual methods. Sequence *10_02* features a dark tunnel passage where visual methods rely solely on faraway features extracted from a reduced size window, as seen in Figures 11a and 11b. This passage affects camera-based odometry techniques, which can be observed in Figure 9c, as all visual methods start drifting in the final left turn because of the erroneous forward self-motion perception during the tunnel traversal that occurs moments before the curve. Consequently, visual techniques obtained a translation error almost six times larger than the point cloud-based alternatives evaluated in sequence *10_02*.

(a)

(b)

FIGURE 11: Tunnel passages in sequences *10_01* and *10_02* [3]. The absence of light inhibits the extraction of good visual features to track, resulting in an altered ego-motion perception.

3) Dynamic objects

Sequence *05_01* has a few large-scale visual occlusion moments (Figure 12). This type of scenario is especially challenging for purely visual techniques since the areas of the field of view occupied by these objects are considerably large, making it difficult to extract quality features and even hindering the detection of moving objects due to the aperture problem. Figure 13 demonstrates the deviation effect on visual methods when a large truck crosses in front of the

TABLE 5: Translational error (meters) of evaluated odometry techniques in KITTI-360 selected sequences.

Methods	Metrics	Sequences										
		00_01	03_01	03_02	04_01	04_02	05_01	07_01	07_02	10_01	10_02	Mean
ORB-SLAM2 [12]△	t_e (m)	**3.10**	2.80	0.28	1.14	0.93	5.98	7.51	115.03	1.68	2.61	13.28
	t_seq (m)	2.31	1.91	0.23	0.70	0.58	5.01	5.52	34.81	1.25	1.44	4.97
LIBVISO2 [14]△	t_e (m)	-	3.53	0.30	1.03	0.71	8.37	18.83	5.58	2.74	4.67	4.69
	t_seq (m)	-	1.95	0.22	0.48	0.52	5.92	12.49	2.58	2.04	2.58	2.95
SOFT [16]△	t_e (m)	8.01	2.53	0.67	2.10	89.72	5.10	39.50	6.68	3.99	7.76	15.46
	t_seq (m)	2.83	2.22	0.62	1.26	29.35	3.33	27.41	5.22	2.95	4.86	7.48
SOFT-SLAM [17]△	t_e (m)	16.20	2.47	**0.13**	1.54	2.01	13.16	26.53	6.90	2.14	3.19	8.30
	t_seq (m)	1.87	2.32	**0.14**	1.38	1.08	7.60	18.73	4.89	1.99	2.71	4.71
Ramezani *et al.* [81]△	t_e (m)	13.12	3.40	0.19	0.84	2.32	2.77	26.92	6.58	0.84	1.22	5.79
	t_seq (m)	2.99	2.22	0.17	0.53	1.37	1.90	23.48	6.16	0.96	1.11	3.97
Ramezani *et al.* w/ IMU [81]△◊	t_e (m)	5.07	1.87	0.20	0.87	1.89	1.08	**2.34**	1.61	0.77	1.20	**1.70**
	t_seq (m)	**0.72**	1.36	0.16	0.57	1.24	0.95	**2.49**	1.85	0.71	1.03	**1.11**
Zhao *et al.* [35]△□	t_e (m)	157.36	49.04	4.33	9.10	7.55	11.54	49.44	29.87	19.99	38.82	36.07
	t_seq (m)	26.51	42.17	4.76	6.88	6.00	11.20	42.12	21.60	16.53	30.68	20.69
SC-SfMLearner [83]△□	t_e (m)	186.69	53.16	4.04	17.98	40.11	34.82	112.12	48.95	17.22	18.41	51.65
	t_seq (m)	36.65	54.55	4.13	15.37	27.22	33.41	140.21	40.02	15.26	13.84	36.25
MULLS [43]○	t_e (m)	33.99	1.56	1.18	1.55	3.14	1.31	11.41	1.25	0.40	0.86	5.38
	t_seq (m)	9.33	1.30	0.98	0.88	1.22	0.94	19.43	0.95	0.44	0.66	3.38
CT-ICP [42]○	t_e (m)	4.33	**0.42**	0.72	1.02	**0.34**	**0.30**	9.43	**1.13**	0.37	0.79	1.80
	t_seq (m)	1.15	**0.19**	0.66	0.46	**0.29**	**0.28**	8.24	**0.92**	0.32	0.92	1.29
F-LOAM [47]○	t_e (m)	13.19	1.11	0.47	**0.60**	3.33	0.66	16.76	40.12	**0.17**	**0.53**	7.16
	t_seq (m)	4.89	0.78	0.54	**0.44**	1.56	0.59	16.13	54.66	**0.19**	**0.47**	7.40
ISC-LOAM [48]○	t_e (m)	13.04	1.10	0.45	**0.60**	2.62	0.67	16.76	40.12	**0.17**	0.60	7.01
	t_seq (m)	4.94	0.71	0.52	**0.44**	1.23	0.52	16.13	54.66	**0.19**	0.59	7.33

Legend: △-Visual, ◯-Point cloud, ◊-Fusion, □-Deep Learning,

agent while it is stationary. Similarly, the final section of the graph in Figure 9g demonstrates the drift that has been accumulated due to visual occlusions. From Table 5 it can be seen that LiDAR-based methods perform more accurately in *05_01*, with a ten times smaller translational error, either because of the 360° coverage of the scene or due to specific dynamic object rejection methods included in the algorithms.

(a)

(b)

FIGURE 12: Visual occlusions caused by two large trucks in sequence 05_01. These scenarios cause severe limitations, especially on visual techniques, which cause deviations in the odometry estimation [3].

Sequence *07_02* contains several start-and-stop moments with the presence of multiple vehicles with similar velocities to the agent, Figure 14. Analogously to sequence *05_01*, multiple dynamic objects disturb the visual algorithms, as numerous cars start moving while the agent remains stationary, causing a false sensation of self-motion. ORB-SLAM2 (Figure 15) composes an evident example of this effect as the translation errors $\overline{t_e}$ and t_{seq} are respectively 8.7 and 7.0 times higher than the average for this technique. In addition, almost all other visual methods achieved a below-average performance too. LiDAR methods show increased robustness to these occurrences, as shown in the results from Table 5. Once more, concerning moving objects in the scene, the integration of the IMU can add a confidence measure to pose estimates, as in these cases, there would be diverging information coming from both sensors. Looking at both versions of Ramezani *et al.* (with and without IMU) in sequence *07_02*, one can witness the referred impact of the IMU, as the average translation drift drops from 6.58 meters to 1.61 meters.

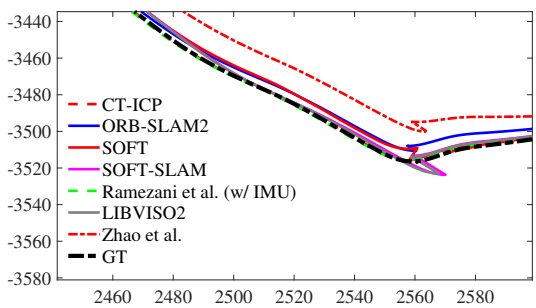


FIGURE 13: Pseudo-motion effect of moving objects on visual methods in sequence 05_01. The trajectories present a significant discontinuity caused by a large-scale occlusion of a moving truck. Axis units in meters.

FIGURE 14: Start and stop moment with multiple moving vehicles in the surrounding area. As the other vehicles move, while the sensing vehicle is stationary, they cause a false sensation of motion [3].

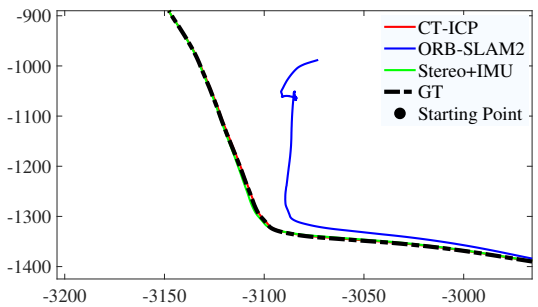


FIGURE 15: Drift effect on ORB-SLAM2 trajectory caused by tracking features belonging to a moving truck in sequence 07_02. Axis units in meters.

B. GENERAL PERFORMANCE COMPARISON

The results from the benchmarking reinforce the prevalence of LiDAR-based methods in a generalized perspective. As can be seen from Table 5, the trajectory errors of LiDAR techniques are consistently lower than the remaining methods when there are no apparent signs of degeneration, as in *07_01* and *07_02*. This indicates that the estimates are not only accurate but also precise. Note that the evaluated error is calculated over the integrated trajectory; thus, a small number of estimates with significant deviations or multiple with minor deviations, would cause the trajectory error to propagate through the entire sequence, gradually or not. In fact, it is noticeable that CT-ICP and F-LOAM/ISC-LOAM dominate in most of the sequences, the last two being close versions of the same method. While purely visual techniques, excluding learning-based, hold an average of 9.5 meters in average translation error, point cloud-based techniques present a lower drift of 5.4 meters. In point cloud odometry, the surrounding 3D structures are rendered directly, as opposed to VO, in which the input is a 2D projection of the scene, subject to pixel discretization and consequently loss of precision. Moreover, while visual techniques' perception is limited to a small fraction of the involving space, common LiDARs have a 360º horizontal coverage. This serves the purposes of LiDAR-based techniques since motion estimation benefits from dispersed points in space while being more resilient to outliers. These direct comparisons help explain why point cloud methods are usually more precise and robust. Moreover, unlike the original KITTI dataset, the point clouds are not undistorted, so algorithms like MULLS, for instance, deliver poorer performances than usual.

Concerning long sequences, such as *00_01* and *07_01* (Figures 9b and 9i), and regardless of the sensing modality of the method, the accumulation of drift along the trajectory will typically occur, as odometry is an integrative process. For this reason, it is always worth relying on techniques that correct the trajectory sporadically, such as mapping and loop closure, as in SLAM algorithms, or the integration of precise GPS measurements. Alternatively, combining multiple sensors, as in Ramezani *et al.* [81], provides redundancy in self-localization, which can avoid the drift tendency compared to when there is only a single sensing reference. The combination of exteroceptive (camera, LiDAR, RADAR, etc.) and proprioceptive sensors (IMU), as in the last example, is especially useful in terms of robustness. This is because the nature of errors that affect these sensors is very distinct, and they will not very likely be affected by the same conditions of the environment.

Moreover, although the work of Ramezani *et al.* [81] outperforms the remaining algorithms in only two sequences, it comes out on top regarding the mean error across all sequences, surpassing the top-performer CT-ICP by roughly 6% in terms of an average translation error. This result indicates that the inclusion of a second data modality (inertial) allows for a more stable performance, which despite being outperformed in some cases, still performs consistently in all tests, proving its ability to stay on track even in the face of the described challenges

Just like the results shown on the original KITTI dataset, the results presented by the learning-based methods still fall far short of traditional methods. Although the tests performed on the KITTI-360 do not have a significant representation at the level of DL methods, the magnitude of the errors obtained and respective trajectories show that their performance is not yet comparable to the visual methods of classical topology, even when compared to baseline models. From raw data to pose estimations, the visual odometry pipeline reflects a complex problem which Deep Learning end-to-end approaches cannot yet reproduce competitive results.

VII. CURRENT CHALLENGES AND LIMITATIONS OF EGO-MOTION ESTIMATION

This section wraps up all the conclusions arising from the previous ones to characterize the current panorama of research in odometry techniques for autonomous driving. The main limiting factors of visual odometry are also identified, along with some possibilities that may potentiate more accentuated advances in this area.

The analysis conducted in Section V shows that the latest improvements in terms of deviation from the trajectory have been residual, regardless of the type of approach taken. In fact, LOAM [44] for example, is eight years old and still performs comparably to its most recent alternatives. It is also noticeable that the increasing complexity in recent works is not being followed by performance at the same rate. This

phenomenon may point out that conventional knowledge-based approaches start showing signs of maturity and may be converging to a development plateau. The two main barriers to the evolution of these techniques are closely related to the high complexity and unpredictability of real environments. On the one hand, odometry systems need to be as robust as possible to all kinds of adverse conditions, such as lighting and weather, which are directly associated with the physical sensing of the surroundings. On the other, the algorithms should have the maximum abstraction possible so that perception, at the software level, can withstand every kind of scenario it may encounter, whether wide open or crowded areas, with many dynamic objects or vegetation, for example. Therefore, this is a joint generalization problem in the physical and software domains.

From a more specific point of view, the most relevant hurdles related to visual and point cloud-based odometry can fit into three categories: scene conditions, computational cost, and dynamic objects.

A. SCENE CONDITIONS

Vision-based systems are highly susceptible to the environment's visual appearance. Visual odometry, more specifically feature-based VO, tends to underperform when the surrounding environment lacks a relevant number of high-quality features that can be tracked in subsequent frames. This can happen during a diverse set of scene conditions, in the nighttime when environment visibility is very low, or in adverse conditions such as heavy rain or fog. Scenarios like a desert or open field surrounded roads also tend to impact these methods because of the featureless nature of such environments. Lighting variations can also impose additional difficulties, especially in the case of direct visual odometry, as high brightness variations, like the entrance of a dark tunnel, might have a significant impact. This happens because the photoconsistency assumption, which assumes a constant brightness across sequential frames used in this type of approach, does not work under these conditions. LiDAR-based techniques, in turn, are not affected by lighting conditions. On the other hand, LiDAR scans are highly affected by weather and atmospheric conditions. Carballo *et al.* [84] tested twelve different LiDAR models in adverse weather conditions like heavy fog and rain, proving several limitations. For example, in heavy rain, the laser beams reflected on raindrops create "rain pillars" in the scans that constitute a high noise level in the point clouds, as shown in Figure 16. As mentioned before, the laser beams' wavelength can impact the sensor's performance in adverse conditions [64]. Laser-based sensors are also heavily affected by reflectivity, failing to detect objects with low reflective properties, such as glass. Like VO, these methods also tend to fail in environments with no apparent salient cues, such as country roads surrounded by broad open fields. These limitations are corroborated by the evaluation in the KITTI-360 dataset, where LiDAR-based methods showed signs of degradation in performance. On the other hand, as expected, these techniques did not report any performance issues when faced with illumination variations. The same cannot be said for visual odometry techniques.

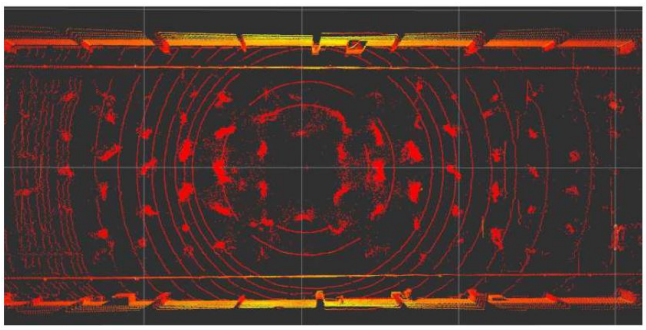


FIGURE 16: LiDAR scan (top view) taken under heavy rain. The lasers reflect on rain drops creating "rain pillars" [84].

B. COMPUTATIONAL COST

Limitations in terms of computing power and time should also be addressed. This is especially important in the context of autonomous driving, where the estimates should be provided at a high enough rate to meet the demanding time constraints associated with autonomous vehicles. Many authors use high-performance hardware to develop their works, presenting a problem because mobile robots are normally equipped with lower-grade hardware. This equipment might be unable to run the algorithms at the required frequency for real-time operation. In VO, computing times are essential if the autonomous vehicle has several cameras, especially if the captured images are high resolution, which can make the amount of data to process unmanageable. One way to reduce the computational burden involves carefully selecting features instead of working with the entire raw images. Nonetheless, feature extraction/matching and outlier rejection tasks may also consume a considerable amount of time. In point cloud-based techniques, the same problems persist, with the added cost related to the unordered nature of point clouds, whose pre-processing operations can be time costly. One way to order and store point clouds efficiently is by using octrees or K-D trees, which allow fast multidimensional searches.

C. DYNAMIC OBJECTS

Dynamic object detection is of great interest while discussing visual and laser odometry, especially in autonomous driving, as this type of object is widespread, i.e., other moving cars, pedestrians, and cyclists. While some authors take dynamic objects into account, most do not. This means that it is assumed that the world is static, which has significant implications when estimating the ego-motion of a vehicle. Dynamic objects, when not accounted for, introduce errors in trajectory estimation. If moving objects are considered static, the odometry computation will be based on wrong assumptions, and the pose estimation will not be accurate. It is also important to note that detecting dynamic objects can be especially complicated in the area of autonomous driving,

as the sensor is mounted on a moving vehicle. Therefore the detected motion has two components: the ego-motion (movement of the sensor) and the object's own motion, both related to the same fixed frame. One way to distinguish these types of motion is to complement odometry algorithms with additional sensors such as IMUs or GPS, to infer the ego-motion and then extrapolate the detected object's movement by subtraction. The evaluation done on the challenging scenarios shows a significant decrease in accuracy when VO methods are faced with highly dynamic environments, except for the camera and IMU fusion method [81]. LiDAR-based methods also tend to be more robust in this type of situation. This is the case because VO algorithms tend to extract features from moving objects (moving objects usually have salient visual features) if there are no proper masking operations, as shown in Figure 17.

FIGURE 17: ORB-SLAM2 extraction of features from moving vehicles. This causes an altered ego-motion perception which leads to incorrect odometry estimates.

Motion analysis techniques, such as optical and scene flow, are typical approaches to infer the dynamics of objects in the scene [85]. It is possible to implement methods for moving object segmentation using the flow fields inferred from optical flow algorithms. For example, in their works, Pinto *et al.* [86], [87] propose various segmentation approaches to segment dynamic objects from a moving robotic platform. The same concepts could be extended and used in autonomous driving. Furthermore, with the recent advancements in optical flow estimation using Deep Learning approaches such as Flownet [88] and RAFT [89], dense flow estimation can be calculated in an accurate and timely manner. Some of the exposed methods in the above sections, such as ClusterVo, SuMa++, and TVL-SLAM, already consider this problem as they employ different techniques such as semantic segmentation or reprojection strategies to filter out the unwanted objects. While not directly related to the odometry task, several authors propose methods for detecting and segmenting dynamic objects in the context of autonomous driving. One example is the work of Chen *et al.* [90] that creates residual images from the LiDAR scans and feeds them into regular point cloud segmentation networks to identify the moving objects. Pfreundschuh *et al.* [91] propose a method to identify dynamic objects with an offline algorithm and then use the labeled data to train a neural network capable of detecting moving obstacles in real-time. FuseMODNet [92] is an example of a multi-modal approach that uses cameras and LiDARs to detect moving obstacles in low-light scenarios.

D. RESEARCH OPPORTUNITIES

Multi-modal architectures have been demonstrated to provide increased robustness in situations where one sensor compensates for the failure of another. Nevertheless, the further development of multi-modal approaches is still needed and justified because there is a strong correlation between the representation of the environment from the agent's perspective and the respective results, which tend to be better, the more complete the representation is. Furthermore, a diverse and complete set of collected data always provides better insights and reliability. For these reasons, part of the limitations of visual odometry algorithms, especially concerning environment sensing, can be mitigated by developing systems that integrate multiple sensors and thus generate more accurate and feasible estimates.

The problems of complex scenes and generalization, however, do not hold obvious solutions. In practice, and regarding the actual sensing options available, it remains very challenging to formulate a reliable real-world model by hand, i.e. the problem of odometry is affected by numerous factors that are very difficult to detect and especially to generalize. Given the wide variety of scenarios that can be encountered, from urban landscapes with dozens of dynamic objects, to long desert roads with potentially very few landmarks to track, in addition to the limitations of the sensors and associated noise. Consequently, one viable option is to engineer the capacity of Deep Learning to capture the more intricate underlying aspects in data, that conventional methods have so far found difficult.

However, learning-based techniques still remain a long way short of what is needed, at least for autonomous driving scenarios. This conclusion is also supported by the results obtained in the KITTI-360 evaluation in Section VI, although the tests are not extensive in this domain. Also, data availability is still not large enough, and existing DL structures, such as CNNs, which have dominated computer vision in the last years, are ineffective in learning sequential data relations. RNNs, in turn, are able to tackle this problem, but fall short on capturing image features the same way CNNs are good at. Therefore, it is frequent to find CNNs followed by RNNs, as common solutions tend to adapt existing structures rather than designing and tailoring them from scratch. Along this line, as the employment of Deep Learning is highly flexible, Ke Wang *et al*. [25] identify the need to bring new techniques and architectures into visual/laser odometry as a future work opportunity. Furthermore, recent works such as D3VO are starting to approximate conventional methods, not through end-to-end architectures, in which the learning process may be too complex, but through learning-based sub-modules that aim to complement the system chain, such as depth prediction module, for instance. Another proposal that seems unattended is the integration of deep-learning algorithms into multi-modal architectures in an effort to leverage the aforementioned advantages of both. Although new approaches are emerging, the panorama of learning-based odometry is still in a very premature state, and thus needs to progress to a more

advanced and suitable state for use in real-world cases, as it is in other areas already.

VIII. CONCLUSIONS

This paper introduced some of the fundamental elements of visual and point cloud-based odometry, as well as an extensive survey of the current state-of-the-art. The best-performing techniques were discussed in light of the results obtained in the KITTI dataset. Furthermore, a representative set of openly available methods, including visual, point cloud, multi-modal and learning-based, were evaluated in a series of challenging scenarios from the KITTI-360 dataset. The results pointed out the predominance of point cloud-based techniques concerning the trajectory translational drift. In particular, CT-ICP [42] achieved an error almost four times smaller than the rest of the assessed algorithms and three times smaller than the other LiDAR-based approaches. The benefits of multi-modal architectures were also validated by the consistency presented in the results obtained by a camera and IMU fusion from Ramezani *et al*. [81]. The outcomes showed no signs of degeneration in none of the evaluated scenarios as the different modalities were able to mitigate the drawbacks of one another efficiently. The reported results on the original KITTI dataset also support these conclusions. The high complexity and variability of the scenes, the weather, the lighting conditions, the dynamic obstacles, and the computational costs were indexed as the most limiting factors in the progress of odometry algorithms, as we identify a current development plateau of traditional methods. This paper reinforces the increased robustness of multi-modal methods as part of the solution, and the necessity of studying and developing better Deep Learning-based solutions to exploit the data-driven modeling capabilities of these approaches, as the current ones are still not competitive. Thereupon, joining deep-learning algorithms and sensor fusion may also be a promising breakthrough in research as it remains slightly unexplored.

REFERENCES

[1] D. Margaria, E. Falletti, and T. Acarman, "The need for gnss position integrity and authentication in its: Conceptual and practical limitations in urban contexts," in 2014 IEEE Intelligent Vehicles Symposium Proceedings, 2014, pp. 1384–1389.

[2] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," IEEE robotics & automation magazine, vol. 18, no. 4, pp. 80–92, 2011.

[3] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," arXiv:2109.13410, 2021.

[4] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," IEEE Robotics & Automation Magazine, vol. 19, no. 2, pp. 78–90, 2012.

[5] Y. Alkendi, L. Seneviratne, and Y. Zweiri, "State of the art in vision-based localization techniques for autonomous navigation systems," IEEE Access, vol. 9, pp. 76 847–76 874, 2021.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in European conference on computer vision. Springer, 2006, pp. 404–417.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in 2011 International Conference on Computer Vision, 2011, pp. 2564–2571.

[9] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in 2011 International Conference on Computer Vision, 2011, pp. 2548–2555.

[10] S. A. Mohamed, M.-H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems," IEEE Access, vol. 7, pp. 97 466–97 486, 2019.

[11] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in International workshop on vision algorithms. Springer, 1999, pp. 298–372.

[12] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," IEEE transactions on robotics, vol. 33, no. 5, pp. 1255–1262, 2017.

[13] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163, 2015.

[14] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in 2011 IEEE Intelligent Vehicles Symposium (IV), 2011, pp. 963–968.

[15] G. Welch, G. Bishop et al., "An introduction to the kalman filter," 1995.

[16] I. Cvišić and I. Petrović, "Stereo odometry based on careful feature selection and tracking," in 2015 European Conference on Mobile Robots (ECMR), 2015, pp. 1–6.

[17] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, "Soft-slam: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," Journal of field robotics, vol. 35, no. 4, pp. 578–595, 2018.

[18] I. Cvišić, I. Marković, and I. Petrović, "Recalibrating the kitti dataset camera setup for improved odometry accuracy," in 2021 European Conference on Mobile Robots (ECMR), 2021, pp. 1–6.

[19] D. Nistér, "An efficient solution to the five-point relative pose problem," IEEE transactions on pattern analysis and machine intelligence, vol. 26, no. 6, pp. 756–770, 2004.

[20] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in CVPR 2011. IEEE, 2011, pp. 2969–2976.

[21] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.

[22] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361.

[23] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in European conference on computer vision. Springer, 2014, pp. 834–849.

[24] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 3, pp. 611–625, 2017.

[25] K. Wang, S. Ma, J. Chen, F. Ren, and J. Lu, "Approaches challenges and applications for deep visual odometry toward to complicated and emerging areas," IEEE Transactions on Cognitive and Developmental Systems, early access. doi:10.1109/TCDS.2020.3038898.

[26] S. Poddar, R. Kottath, and V. Karar, "Motion estimation made easy: Evolution and trends in visual odometry," in Recent Advances in Computer Vision. Springer, 2019, pp. 305–331.

[27] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 817–833.

[28] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 2043–2050.

[29] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers, "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1281–1292.

[30] S. Wang, R. Clark, N. Trigoni, and H. Wen, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," The International Journal of Robotics Research, vol. 37, no. 4-5, pp. 513–542, 2018.

[31] G. Zhai, L. Liu, L. Zhang, Y. Liu, and Y. Jiang, "Poseconvgru: A monocular approach for visual ego-motion estimation by learning," Pattern Recognition, vol. 102, p. 107187, 2020.

[32] R. Zhu, M. Yang, W. Liu, R. Song, B. Yan, and Z. Xiao, "Deepavo: Efficient pose refining with feature distilling for deep visual odometry," Neurocomputing, vol. 467, pp. 22–35, 2022.

[33] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8934–8943.

[34] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.

[35] W. Zhao, S. Liu, Y. Shu, and Y.-J. Liu, "Towards better generalization: Joint depth-pose learning without posenet," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9151–9161.

[36] J. Huang, S. Yang, T.-J. Mu, and S.-M. Hu, "Clustervo: Clustering moving instances and estimating visual odometry for self and surroundings," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2168–2177.

[37] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," in 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 6939–6946.

[38] N. Radwan, A. Valada, and W. Burgard, "Vlocnet++: Deep multitask learning for semantic visual localization and odometry," IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 4407–4414, 2018.

[39] N. Jonnavithula, Y. Lyu, and Z. Zhang, "Lidar odometry methodologies for autonomous driving: A survey," arXiv:2109.06120, 2021.

[40] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," International journal of computer vision, vol. 13, no. 2, pp. 119–152, 1994.

[41] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in Sensor fusion IV: control paradigms and data structures, vol. 1611. Spie, 1992, pp. 586–606.

[42] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," arXiv:2109.12979, 2021.

[43] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "Mulls: Versatile lidar slam via multi-metric linear least square," arXiv:2102.03771, 2021.

[44] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in Robotics: Science and Systems, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[45] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 4758–4765.

[46] X. Zheng and J. Zhu, "Efficient lidar odometry for autonomous driving," arXiv:2104.10879, 2021.

[47] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam : Fast lidar odometry and mapping," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 4390–4396.

[48] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 2095–2101.

[49] M. Oelsch, M. Karimi, and E. Steinbach, "R-loam: Improving lidar odometry and mapping with point-to-mesh features of a known 3d reference object," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 2068–2075, 2021.

[50] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, "Lo-net: Deep real-time lidar odometry," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8473–8482.

[51] Y. Cho, G. Kim, and A. Kim, "Deeplo: Geometry-aware deep lidar odometry," arXiv:1902.10562, 2019.

[52] C. Zheng, Y. Lyu, M. Li, and Z. Zhang, "Lodonet: A deep neural network with 2d keypoint matching for 3d lidar odometry estimation," in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 2391–2399.

[53] G. Wang, X. Wu, Z. Liu, and H. Wang, "Pwclo-net: Deep lidar odometry in 3d point clouds using hierarchical embedding mask optimization," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15 910–15 919.

[54] G. Chen, B. Wang, X. Wang, H. Deng, B. Wang, and S. Zhang, "Psf-lo: Parameterized semantic features based lidar odometry," in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 5056–5062.

[55] D. Yin, Q. Zhang, J. Liu, X. Liang, Y. Wang, J. Maanpää, H. Ma, J. Hyyppä, and R. Chen, "Cae-lo: Lidar odometry leveraging fully unsupervised convolutional auto-encoder for interest point detection and feature description," arXiv:2001.01354, 2020.

[56] Z. Li and N. Wang, "Dmlo: Deep matching lidar odometry," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 6010–6017.

[57] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 4530–4537.

[58] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments." in Robotics: Science and Systems, vol. 2018, 2018.

[59] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet ++: Fast and accurate lidar semantic segmentation," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 4213–4220.

[60] D. J. Yeong, G. Velasco-Hernandez, J. Barry, J. Walsh et al., "Sensor and sensor fusion technology in autonomous vehicles: A review," Sensors, vol. 21, no. 6, p. 2140, 2021.

[61] W. Wang, J. Liu, C. Wang, B. Luo, and C. Zhang, "Dv-loam: Direct visual lidar odometry and mapping," Remote Sensing, vol. 13, no. 16, p. 3340, 2021.

[62] D. Wisth, M. Camurri, S. Das, and M. Fallon, "Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1004–1011, 2021.

[63] M. Kutila, P. Pyykönen, W. Ritter, O. Sawade, and B. Schäufele, "Automotive lidar sensor development scenarios for harsh weather conditions," in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 265–270.

[64] J. Wojtanowski, M. Zygmunt, M. Kaszczuk, Z. Mierczyk, and M. Muzal, "Comparison of 905 nm and 1550 nm semiconductor laser rangefinders' performance deterioration due to adverse environmental conditions," Opto-Electronics Review, vol. 22, no. 3, pp. 183–190, 2014.

[65] Q. Liao, Z. Chen, Y. Liu, Z. Wang, and M. Liu, "Extrinsic calibration of lidar and camera with polygon," in 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2018, pp. 200–205.

[66] J. Domhof, J. F. Kooij, and D. M. Gavrila, "An extrinsic calibration tool for radar, camera and lidar," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 8107–8113.

[67] D. J. Yeong, G. Velasco-Hernandez, J. Barry, J. Walsh et al., "Sensor and sensor fusion technology in autonomous vehicles: A review," Sensors, vol. 21, no. 6, p. 2140, 2021.

[68] J. Peršić, I. Marković, and I. Petrović, "Extrinsic 6dof calibration of a radar–lidar–camera system enhanced by radar cross section estimates evaluation," Robotics and Autonomous Systems, vol. 114, pp. 217–230, 2019.

[69] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "3d lidar–camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization," pp. 452–467, 2012.

[70] J. Beltrán, C. Guindel, F. García et al., "Automatic extrinsic calibration method for lidar and camera sensor setups," IEEE Transactions on Intelligent Transportation Systems, 2022.

[71] J.-K. Huang and J. W. Grizzle, "Improvements to target-based 3d lidar to camera calibration," IEEE Access, vol. 8, pp. 134 101–134 110, 2020.

[72] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1360–1367.

[73] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in 2013 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2013, pp. 473–480.

[74] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 2088–2095.

[75] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2013, pp. 1280–1286.

[76] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes," in 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016, pp. 4304–4311.
[77] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: low-drift, robust, and fast," in 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 2174–2181.
[78] J. Graeter, A. Wilczynski, and M. Lauer, "Limo: Lidar-monocular visual odometry," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 7872–7879.
[79] C.-C. Chou and C.-F. Chou, "Efficient and accurate tightly-coupled visual-lidar slam," IEEE Transactions on Intelligent Transportation Systems, pp. 1–15, 2021.
[80] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," The International Journal of Robotics Research, vol. 38, no. 6, pp. 642–657, 2019.
[81] M. Ramezani and K. Khoshelham, "Vehicle positioning in gnss-deprived urban areas by stereo visual-inertial odometry," IEEE Transactions on Intelligent Vehicles, vol. 3, no. 2, pp. 208–217, 2018.
[82] A. R. Gaspar, A. Nunes, A. M. Pinto, and A. Matos, "Urban@ cras dataset: Benchmarking of visual odometry and slam techniques," Robotics and Autonomous Systems, vol. 109, pp. 59–67, 2018.
[83] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," Advances in neural information processing systems, vol. 32, 2019.
[84] A. Carballo, J. Lambert, A. Monrroy, D. Wong, P. Narksri, Y. Kitsukawa, E. Takeuchi, S. Kato, and K. Takeda, "Libre: The multiple 3d lidar dataset," in 2020 IEEE Intelligent Vehicles Symposium (IV), 2020, pp. 1094–1101.
[85] A. M. Pinto, A. P. Moreira, M. V. Correia, and P. G. Costa, "A flow-based motion perception technique for an autonomous robot system," Journal of Intelligent & Robotic Systems, vol. 75, no. 3, pp. 475–492, 2014.
[86] A. M. Pinto, M. V. Correia, A. P. Moreira, and P. G. Costa, "Unsupervised flow-based motion analysis for an autonomous moving system," Image and Vision Computing, vol. 32, no. 6-7, pp. 391–404, 2014.
[87] A. M. Pinto, P. G. Costa, M. V. Correia, A. C. Matos, and A. P. Moreira, "Visual motion perception for mobile robots through dense optical flow fields," Robotics and Autonomous Systems, vol. 87, pp. 1–14, 2017.
[88] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), December 2015.
[89] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in European conference on computer vision. Springer, 2020, pp. 402–419.
[90] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss, "Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data," arXiv:2105.08971, 2021.
[91] P. Pfreundschuh, H. F. C. Hendrikx, V. Reijgwart, R. Dubé, R. Siegwart, and A. Cramariuc, "Dynamic object aware lidar slam based on automatic generation of training data," arXiv:2104.03657, 2021.
[92] H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu, and S. Yogamani, "Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Oct 2019.

LUCAS AGOSTINHO received his B.S. degree in electrical and computer engineering from the University of Porto, Portugal, in 2020. He is currently pursuing an M.S. degree in robotics and autonomous systems from the same institute, and researching in the field of multi-modal perception for autonomous vehicles. His interests cover mobile robotics, computer vision, and industrial automation.

NUNO RICARDO obtained his B.S. degree in electrical and computer engineering from the University of Porto, Portugal, in 2020. He is currently pursuing an M.S. degree in robotics and autonomous systems from the same institute, while researching in the field of perception and autonomous vehicles. His main interest topics are computer vision, automation and robotics.

MARIA PEREIRA received the M.Sc. degree in electrical and computer engineering from the Faculty of Engineering, University of Porto (FEUP), Portugal, in 2020. She is currently pursuing the Ph.D. degree in electrical and computer engineering, while conducting research with the Centre for Robotics and Autonomous Systems, INESC TEC. Her main research interests include maritime robotics, Deep Learning, and perception systems.

ANTOINE HIOLLE received his M.S. degree in software engineering from CY Tech, Paris, France in 2005 and his PhD degree in computer and information science in 2015 from the University of Hertforshire, United Kingdom. From 2010 to 2015, he was a Research Fellow with the University of Hertforshire, working on adaptive robot control systems. He was also a Visiting Lecturer in the same Institute from 2008 to 2011. Currently, he holds the position of Product Owner in Bosch Car Multimedia Portugal.

ANDRY PINTO received the Ph.D. degree in electrical and computer engineering from the Faculty of Engineering, University of Porto, Portugal, in 2014. He is currently an Assistant Professor with the Faculty of Engineering, University of Porto, and a Senior Researcher with the Centre for Robotics and Autonomous Systems, INESC TEC. He is the Principal Investigator of national and international research and development projects related to robotic-based operation and maintenance (OM) activities for offshore infrastructures. His main research interests include multi-domain perception, underwater imaging, artificial intelligence, and mobile robotics.